Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations 1. Thesis and Dissertation Collection, all items

1984

PROJMNG FORTRAN : an interactive computer program for use with the Defense Management Simulation exercise.

Schultz, George

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/19512

Downloaded from NPS Archive: Calhoun

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

http://www.nps.edu/library

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CALIFORNIA 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

PROJMNG FORTRAN:
AN INTERACTIVE COMPUTER PROGRAM FOR USE WITH THE DEFENSE MANAGEMENT SIMULATION EXERCISE

by

George W. Schultz

March, 1984

Thesis Advisor: M. B. Kline

Approved for public release, distribution unlimited

T215689

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*
PROJMNG FORTRAN: An Interactive Computer Program for Use with the Defense Management Simulation Exercise

5. TYPE OF REPORT & PERIOD COVERED
Master's Thesis
March, 1984

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
George W. Schultz

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93943

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Postgraduate School
Monterey, California 93943

12. REPORT DATE
March, 1984

13. NUMBER OF PAGES
323

14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*

15. SECURITY CLASS. *(of this report)*
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*
Approved for public release, distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Project Management Training
Acquisition Management Training
Defense Management Simulation
PROJMNG
Contract Negotiation Training
Contract Negotiation
Computer Aided Training

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Contract Negotiation Package (CNP), the supporting computer program for the Defense Management Simulation (DMS), is revised and embedded into a program which makes it user-friendly, and which provides sensitivity analysis capability to it. The program includes a plotting function for the sensitivity analysis. Exercise records are established for review of contracting team performance. Database files are generated which permit teams to submit reports, which provide a baseline for subsequent game sessions, and which

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

permit monitor evaluation of team progress.

The text provides a description for the operation of both CNP and PROJMNG. It documents the new program, PROJMNG.

Appendices include the Fortran code, and Conversational Monitor System (CMS) executive machine language programs for the new programs operation. It contains instruction manuals which depict operation for both programs.

Approved for public release; distribution unlimited.

PROJMNG FORTRAN:
An Interactive Computer Program for Use with the Defense Management Simulation Exercise

by

George W. Schultz
Lieutenant Commander, United States Navy
B.S.E.E., University of New Mexico, 1971

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(COMMAND, CONTROL AND COMMUNICATIONS)

from the

NAVAL POSTGRADUATE SCHOOL
March 1984

52645 3
.1

ABSTRACT

Contract Negotiation Package (CNP), the supporting computer program for the Defense Management Simulation (DMS), is revised and embedded into a program which makes it user-friendly, and which provides sensitivity analysis capability to it. The program includes a plotting function for the sensitivity analysis. Exercise records are established for review of contracting team performance. Database files are generated which permit teams to submit reports, which provide a baseline for subsequent game sessions, and which permit monitor evaluation of team progress.

The text provides a description for the operation of both CNP and PROJMNG. It documents the new program, PROJMNG.

Appendices include the Fortran code, and Conversational Monitor System (CMS) executive machine language programs for the new programs operation. It contains instruction manuals which depict operation for both programs.

TABLE OF CONTENTS

I. INTRODUCTION 12

- A. HISTORICAL BACKGROUND 12
- B. SCENARIO 13
- C. SIMULATION 15
- D. SUMMARY 16

II. CNP OPERATION 18

- A. PROCEDURES 18
- B. BENEFITS 22
- C. DEFICIENCIES 23
- D. MONITOR FUNCTIONS 25
- E. SUMMARY 26

III. PROJMNG OPERATION 27

- A. CNP MODERNIZATION 27
- B. PROJMNG PACKAGE 28
  - 1. Programs 28
  - 2. Data files 29
  - 3. LINK 30
  - 4. PROJMNG EXEC 31
  - 5. PROFILE EXEC 32
- C. DATABASE SECURITY 33
- D. EXIT 34
  - 1. Fatal Errors 35
  - 2. Security Violation 36
  - 3. Menu 36
  - 4. YES/NO Query 36
  - 5. Return to PROJMNG EXEC 36
- E. PROJMNG PROCEDURES 37

1. Instructions 37
2. Link 37
3. PROJMNG EXEC Execution 38
4. Running PROJMNG 40

F. DATABASE CHANGE 41

G. MENU 42

H. REPORT SUBMISSION 44

1. Progress Reports 44
2. Decision Submission 46

I. CALCULATION CORRECTION 51

J. SUMMARY 53

IV. PROJMNG MONITOR OPERATION 54

A. CNP MONITOR UNIQUE FUNCTIONS 54

B. PROJMNG MONITOR UNIQUE FUNCTIONS 56

C. MONITOR UNIQUE SECURITY SYSTEM 56

D. QUERY SUPPRESSION FOR THE MONITOR 57

E. MONITOR'S MAIN MENU 59

1. Change Team Number 59
2. The DP Selection Query 63
3. The Input Selection Menu 63
4. Rerun the Data Calculation 64
5. Do a Sensitivity Analysis 64
6. Printout Cost + Factors 64
7. EXIT 64
8. Flexible Menu 65

F. MONITOR SESSION PROCEDURES 65

G. INITIALIZATION 66

1. PROJMNG EXEC Operation 66
2. Available Files 66
3. Purging DATAFILEs 67

H. BASIC PROGRAM ROUTINES 69

1. Start of PROJMNG's Run 70
2. Summary of Basic Capabilities 74

I. SENSITIVITY ANALYSIS ROUTINES 74
  1. Sensitivity Analysis Basic Menu Options 74
  2. Sensitivity Analysis Table 78
  3. Menu Expansion 78
  4. Sensitivity Analysis Plotting 79
  5. Effectiveness Index 86
  6. Flexible Menu Reset 86
J. SUMMARY 87

V. SUMMARY AND CONCLUSIONS 89
A. SUMMARY 89
B. DMS 89
  1. Benefit 89
  2. Computer Support 90
  3. CNP 90
C. PROJMNG 90
  1. Benefits 90
  2. Shortcomings 91
  3. First Usage 91
  4. Transportability 92
D. DOCUMENTATION 93
  1. CNP 93
  2. PROJMNG 93
  3. Glossary 93
  4. Debugging 94

VI. RECOMMENDATIONS FOR FUTURE CONSIDERATION 95
A. INCREASED EXERCISE SUPPORT 95
  1. Discussion 95
  2. Recommendation 96
B. INCLUDE ALL DECISION POINTS 96
  1. Discussion 96
  2. Recommendation 97

C. DEVELOP A DMS TUTORIAL 97
    1. Discussion 97
    2. Recommendation 97
D. VALIDATE PROJMNG 98
    1. Discussion 98
    2. Recommendation 98
E. REPLACE THE DATA TABLES WITH EQUATIONS 98
    1. Discussion 98
    2. Recommendation 99
F. INCREASE SENSITIVITY ANALYSIS OPTIONS . . . 100
    1. Discussion 100
    2. Recommendation 100
G. IMPROVED GRAPHICS 100
    1. Discussion 100
    2. Recommendation 102
H. PLATE DISPLAYS 102
    1. Discussion 102
    2. Recommendation 103
I. A SCENARIO GENERATING PROGRAM 104
    1. Discussion 104
    2. Recommendation 104
J. USE PROJMNG TO ENCOMPASS OTHER CNP
VARIATIONS 105
    1. Discussion 105
    2. Recommendation 105

APPENDIX A: CONTRACT NEGOTIATION PACKAGE (CNP) 106

APPENDIX B: DMS DATASHEETS 123

APPENDIX C: LINKPROJ EXEC 132

APPENDIX D: PROJMNG EXEC 134

APPENDIX E: PROFILE EXEC FOR PROJMNG 141

APPENDIX F: PROJMNG STUDENT INSTRUCTIONS 143
A. INTRODUCTION 144
B. INSTRUCTIONS 147

APPENDIX G: PROJMNG FORTRAN 193

APPENDIX H: PROJMNG PROGRAM VARIABLES GLOSSARY 312

INITIAL DISTRIBUTION LIST 322

LIST OF TABLES

I. Development Contract Values Summary 20
II. Page Data Strings 21
III. CCST + FACTORS 25
IV. DATABASE PRINTOUT as SELECTION MENU 43
V. Development Contract Summary 45
VI. Proposed Values Table 47
VII. Final Contract Values Table 48
VIII. Accepted Values Table 49
IX. DATAFILE TEAM03 50
X. CCST + FACTORS 55
XI. SELECTION MENU 58
XII. CCST + FACTORS 65
XIII. Selection Menu 72
XIV. Development Contract Summary 76
XV. Sensitivity Analysis Menu 77
XVI. Sensitivity Table 80
XVII. Development Contract Summary 109
XVIII. Page Data Strings 111
XIX. Development Contract Summary 112
XX. Page Data Strings 115
XXI. Page Data Strings 117
XXII. Development Contract Achieved Value Summary . . 118
XXIII. Development Contract Achieved Value Summary . . 121
XXIV. PROJMNG FILEDEFS 144
XXV. PROJMNG FILES 146

LIST OF FIGURES

1.1 Defense Management Simulation Decision Points 14
3.1 MAIN MENU 42
4.1 Monitor's Main Menu 59
4.2 Monitor Mode Menu 60
4.3 MAIN MENU in Mode 1 61
4.4 MAIN MENU in Mode 2 61
4.5 Initial Monitor Display 68
4.6 Monitor Session File Purge Option 69
4.7 PROJMNG Initial Displays 70
4.8 Team Number Responses 71
4.9 Monitor's Main Menu 74
4.10 Flight Tests Demonstration 75
4.11 Monitor's Main Menu in Mode 1 79
4.12 Plot Character Key Definitions 81
4.13 Sensitivity Table 82
4.14 Sensitivity Analysis Percentages Graph 83
4.15 Monitor's Main Menu in Mode 1 84
4.16 Selected Vertical Axis 85
4.17 Selection Menu with Optimized Flight Tests . . . 88

I. INTRODUCTION

A. HISTORICAL BACKGROUND

During the Presidential Administration of John F. Kennedy, Secretary of Defense Robert S. McNamara instituted a major revision for management of defense resources. These procedures were instituted as a revolution to the military procurement system. Among the revisions he instituted were the Planning, Programming and Budgeting System and a management system for acquisition of new major systems.

The impact of these new procedures on the military was to create an immediate need for extensive education of the armed services' acquisition organizations. The personnel in charge of ongoing system procurements, the project managers, needed a means of being educated about the system. New personnel to the field of acquisition also had to have a means of learning about the system.

In response to the services' need, in 1963 the Defense Weapons Systems Management Center developed the Project Management Simulation Exercise (PMSE) to aid in DOD-wide training in project management. The exercise simulated procurement of a missile system to aid the DOD-wide training in PPBS.

In the later 1960's, the Industrial College of the Armed Forces (ICAF) obtained the simulation and revised it into the Defense Management Simulation (DMS). DMS was installed at the Naval Postgraduate School in 1971.

The exercise currently has international application. It is used by Israel, Sweden and Singapore to train their defense personnel in the use of acquisition management systems.

Agencies within the U. S. government currently using DMS or related products for training exercises include:

- Industrial College of the Armed Forces
- Naval Postgraduate School
- U. S. Air Force Academy

It is this wide acceptance which has maintained DMS as an active management training exercise.

B. SCENARIO

The exercise developed by DWSMC in 1963 consisted of a time-sequenced set of decision points in the life cycle of a missile system acquisition (see Figure 1.1). The decision points (DPs) simulate the equivalent functions of these major milestones in the system life cycle. Within a prescribed time period each team of students participating in the exercise reviews the information materials for the specific decision point and submits its decision for recommended action at that point in the game, as would be done in actuality. The objective is to optimize the cost effectiveness for procurement, production and development of the proposed new missile system, the Zebra Missile System.

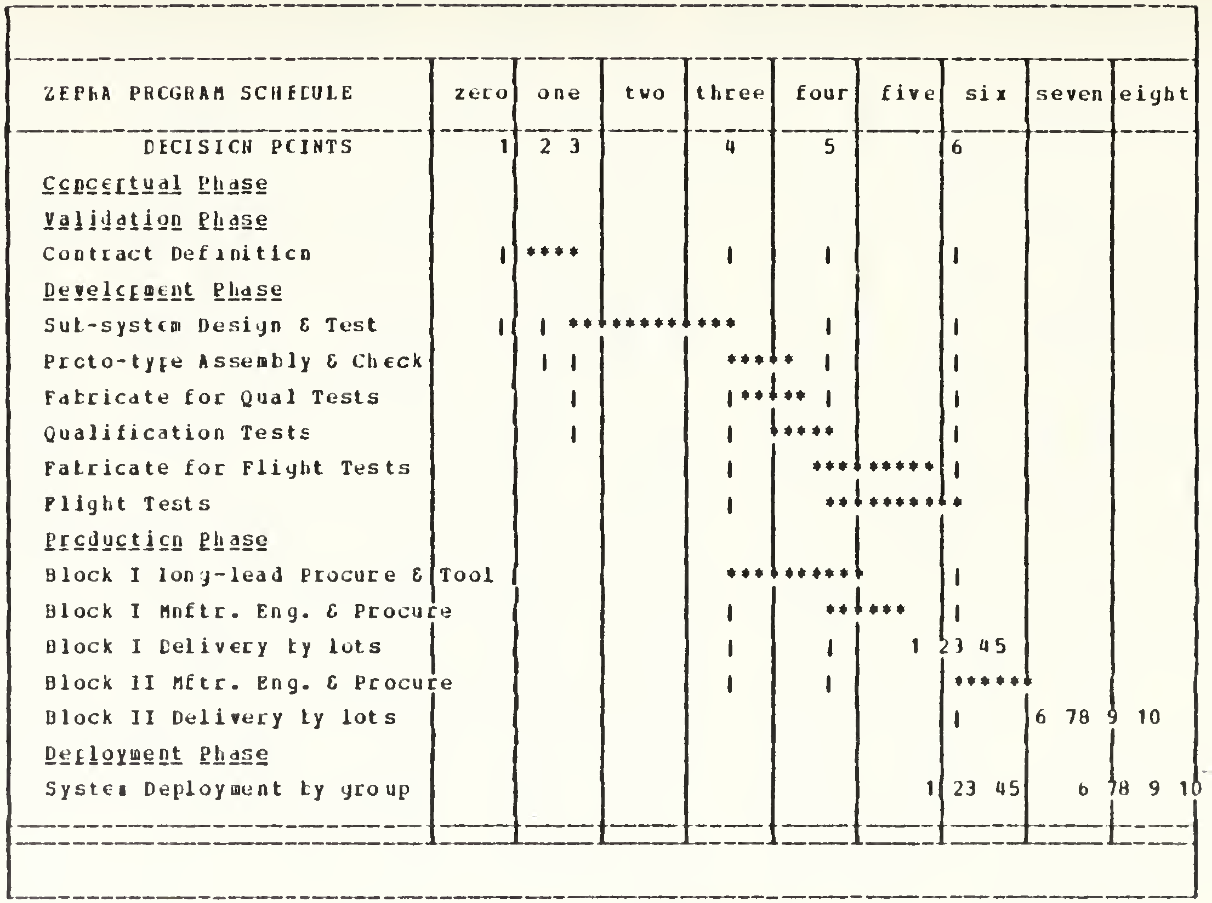


Figure 1.1 Defense Management Simulation Decision Points.

C. SIMULATION

The team recommendations require monitor evaluation. Monitors review the student recommended performance reliabilities, costs, incentives, contractor selection processes, milestone dates and fees. They prepare materials for teams to interact with as they would with the Secretary of Defense and with the contractor. An extensive study of the materials from each team is required by the monitor.

In order for the monitors to not spend as much time on the simulation as the students, the monitors need aid in processing the team data. ICAF developed an Fortran program called Contract Negotiation Package (CNP) to assist the monitor in the evaluation of student team contract negotiations.

The DMS program is more than twenty years old. The program has experienced several modifications and offshoots. Its 1960's programming state-of-the-art remains mostly intact. By 1984 standards of ease of operation and user-friendliness, it is difficult to utilize.

Efforts to modify the program occurred in 1972 and in 1977. The first attempt to upgrade the DMS program in 1972 modified several internal algorithms and produced an offshoot program which provided fewer but more rapid results. In 1977 this version of the program, the Contract Negotiation Package, was again revised. This later revision provided predictions of achieved values to the student teams.

Team access to the program's simulated results grew out of the modernization of management processes. Managers are using the benefits of modern computer systems to improve their own performance. Student team use of the CNP program for the DMS exercise was a logical evolution. CNP use by the teams enables the DMS exercise to reach the level of current

project management technological support in computer-aided acquisition management techniques. The program also remains abreast through student and monitor interest in maintaining the program at the computer support state-of-the-art.

The CNP program available on the IBM-3033 in the W. R. Church Computer Center at the Naval Postgraduate School campus is discussed further in Chapter 2. Its displays reveal CNP's batch program origins. The data input and output formats are batch processing type fields. The lack of interactive capability typifies a program which is batch processed. In its batch processing use, the computer did not have terminals or users with which to interact. Originally CNP had no hardware capable of directly interacting with the users and its design developed accordingly.

Modern menu-driven designs have made user-friendly programming standard practice. As a viable training exercise for acquisition management, the DMS game continues to be valuable. In order to maintain its wide acceptance and expand its efficient utilization, a major revision to the program is needed.

The following chapters discuss a package of programs designed to give CNP an improved interactive program capability.

D. SUMMARY

The DMS training program incorporates an exercise which simulates a project management scenario. DMS is a successful training device which has been in use for 20 years. In support of DMS, the Fortran computer program CNP was developed to provide computer-aided assistance for monitor evaluation of team performance. Later, CNP was released to the student teams in a format which provides them with simulated performance data. It enables team

interaction with the computer. In so interacting, the team gains experience using the computer as an analysis tool and to perform sensitivity analysis ("what if questions") on major project performance and incentive parameters. As a result of CNP's age, its design did not permit adequate interactive capability for easy operation. In order to resolve the need for greater user-friendliness in CNP, this thesis develops, studies and reports on a CNP program revision entitled PRCJMNG FORTRAN.

II. CNP OPERATION

A. PROCEDURES

Proficient use of the Contract Negotiation Program (CNP) depends upon the user's familiarity with both the computer facilities and with the instructions in Appendix A. The "Defense Management Simulation Instructions for Using CNP" presents computer terminal display lines as seen during a terminal session. In following sequentially through the instructions, difficulties are apparent with the program both in user-friendliness and in design. Appendix A demonstrates a session in CNP, and pertains to the following discussion of that program.

The procedures for accessing CNP require the operator to link to the monitor's class disk, then to access the disk, and finally to execute CNP. Accessing the program in this manner normally requires at least a conceptual knowledge of computer command and response terminology. In the event of an accessing malfunction, or if the monitor is using the CNP program's class disk and prevents read-only execution of the disk files, the user may be confronted with unfamiliar terminology. He would need more than a casual knowledge of computer operation to proceed into CNP.

The first function performed by CNP is to input administrative data. Included in the administrative data are the team number, the decision point (DP), and the DP pages for which data are to be entered. Complications with the program quickly arise with these procedures. The data must be input in the exact format prescribed on the instruction sheet. Too many digits, or too many spaces in the data result in erroneous output, and/or error statements. Each

page's data string requires confirmation. This becomes laborious in repeated runs of the program, or repeated executions of the calculations for a Summary Table, Table I.

Frequently explanation is required for the procedure of binary coded response to the query for pages of data to be entered. Coding the page selection for data input can be confusing. Each of the four pages requiring input data must be indicated by '1'(on) or '0'(off) in the four digit code string. Indicating the page number in Arabic characters would be a more commonly expected procedure. Again, an error in entry is irrevocable. The change pages cannot be rescinded once the binary code has been entered.

Having completed the administrative inputs, the team proceeds with providing its proposed DP data. Prior to running CNP, team participants are to have used the DMS Decision Point handouts (see Appendix B) to determine and analyze their tentative DP positions. These results are listed as data lines in the DMS Decision Point Sheets, (Appendix B). Each character in the page's data lines requires precise replication when placed in the page data string input to CNP. The program types the page's data back onto the screen and asks for confirmation. Through relooping to the query for pages of input, the program does allow revision of data. However, each data item to be changed requires an entire page change. The team is confronted with reentering the data string each time. This procedure has proven to be tedious. It frequently results in teams redoing data strings several times to get them correct before a calculation of achieved values can be made. Successive loops through the program have the added check feature of receiving a four line printout of the page data for confirmation. The redundant queries for confirmation of data values have benefit in reducing erroneous data calculations, (see Table II).

Based on the author's experience, about one out of four occasions through the four line recap of data pages, Table II , the team finds an error.

TABLE I

Development Contract Values Summary

```
DP-3 *** DEVELOPMENT CONTRACT SUMMARY *** TEAM   1

                    INCENTIVE PROVISIONS          INCENTIVE
                                                ACHIEVEMENTS

  INCENTIVE        WORST      BEST    MAX FEE     ACHIEVED
    AREA           VALUE      VALUE   ALLOWED      VALUE

DEV. COST        $ 53.0M   $ 47.0M     4.0%      $ 48.5M     $
FLT TST COMPL     238 WK    202 WK     3.5%       199 WK     $
RELIABILITY        75.0%     81.0%     3.5%        80.0%     $
ACCURACY          160YDS   140 YDS     4.0%      143 YDS     $

TOTALS                                15.0%                  $

TOTAL CONTRACT PRICE = $ 54.9M
```

The Development Contract Summary Table, Table I , neatly displays the decision point incentive input data as the 'Incentive Provisions'. Only five calculated values are displayed to the team. They are the achieved values for development cost, flight test completion date, reliability, accuracy, and total contract price. Also, displayed on the Summary Table is the computed total of fee percentages obligated. The total fees allocated must be 15%. Allocation of fees to other than 15% results in a warning on the Summary Table. The significance of the Development Contract Summary table and CNP rests in these five achieved values.

TABLE II

Page Data Strings

```
10301        1       1       2      1.75    0.75    0.0

10302    92.00   97.00   98.50   80.00   70.00   25.00    6.00
         3.00    3.00    6.00    17.00

10303    53.00   47.00    4.00  238.00  202.00    3.50   81.00
         75.00    3.50

10304   160.00  140.00    4.00  337.00
```

Compared with the original CNP design, there are missing values from the Summary Table. It does not print out ten additional calculated values: the fee percentages achieved, their fee cost, and the totals of fees earned and of fees cost. These values were suppressed from the original program table due to their computation by incorrect algorithms. They're suppression is evidenced by the dollar signs at the right of the Summary Table.

To reloop through the input of data and receive another estimate of the achieved values, the team answers an unessential query. This question has no application to the teams. The "COSTS+FACTORS(NO)" query provides monitor access to the tables of design performance factors achieved and test costs in the form of unit change cost factors. The team's no('N') response to, or an incorrect response to, this query reloops the program to the top of the main routine. Again, each question: decision point, team, pages, data, yes or no, and confirm must be answered.

It is extremely difficult to exit from the program if the team attempts to stop at any point before the reloop query. At the query "DO YOU WISH TO CONTINUE, Y FOR YES, N

FOR NC", the program provides its only graceful exit. Any other attempts to exit the program must be accomplished by dumping (stopping) the program. Inputs that do not match the data types required for data read-statement formats result in error statements, and with sufficient error occurrence count the program will dump.

B. BENEFITS

The benefits for team utilization of CNP are in its ability to provide some semblance of the achieved values which the main DMS batch program, provides. In so doing, the team uses the computer as a tool with which to view the sensitivity of its incentivization parameters. It can evaluate effects of varying available DMS critical incentive parameters: development cost, flight test completion date, reliability, accuracy and development contract price. This aspect of analysis simulates the real world potential for computers to expedite the computations and as a tool in these assessments. It further provides an environment in which team participation and success depends on understanding the system acquisition concepts that DMS illustrates. In order to expeditiously vary the parameters within the computer the team must have competent knowledge of these DMS concepts.

CNP's best feature appears in its structure. It closely follows the procedures in the DMS Decision Point Handouts,[1] From its inception as an offshoot of the main CMS program, it has inherently followed the decision point sequence and structure of DMS.

---

[1] Industrial College of the Armed Forces, Defense Management Simulation Participant's Manual, Industrial College of the Armed Forces

The wide acceptance of CNP at many institutions can be found in the attributes it brought to DMS. In meeting the objectives of providing a program with the ability to quickly reloop through computations, and the ability to be operated from a terminal, the program fulfills a major need in performing certain sensitivity analyses over the DMS program. Once accessed, the program can be rerun innumerable times without completely reinputting all the data. The data inputs can be accomplished from a terminal. Computer support for DMS on computer terminals has made DMS accessible to more users without adding expense. It has enabled many students to simultaneously utilize DMS as a tool in acquisition management training.

C. DEFICIENCIES

Areas which need improvement in CNP focus on user-friendliness. The first of these is the inability to gracefully exit from the program. Some means must be available to permit the team to stop the program. Currently the program can only be stopped by expending time to finish the session or by generating sufficient syntax errors to force the computer to dump the program. In some cases, sessions have been in sufficient difficulty that the program would not reach the query "DO YOU WISH TO CONTINUE...". For this situation the team is forced to generate error counts which will dump the program. These situations have been exceedingly frustrating for the program users.

A means to change data entries without redoing entire pages is needed. The team must completely redo a DP page in order to change one data item. Changing one item should only require reentering one value. Additionally, data strings from one decision point to another are not readily duplicated for input. Similar strings of data for pages on

DP-3 and DP-4 do not have the same field format. Matching field formats for identical pages in different DP's would enhance the ability of teams to input data strings.

CNP receives all of the input data but cannot retain it between sessions. The team starts from a baseline of no data each time CNP is called. Most significant of the changes needed to modernize CNP is the creation of a database file to retain team data from session to session. A foundation data base of the initial paged input could provide a means of replicating session results, of documenting data inputs, and of proceeding to successive DP levels without laboriously reentering previously studied data.

Excessive recurrences of calculating the Summary Table could be eliminated to save team time in the program. Preventing runs which produce Summary Tables without correct incentive fee percentage totals would reduce wasted program run time. Utilizing data that does not meet the DMS parameters calculates erroneous values, and may provide Summary Tables which mislead the team. Repeatedly running calculations to evaluate wide ranges of parameter values without conducting proper preliminary analyses not only detracts from the timeliness of the DMS studies, it defeats the purpose of DMS training teams to understand the concepts of project management. Teams have occasionally found greater ease in working through the program and produce a Summary Table in order to reach the "DO YOU WISH TO CONTINUE...?" query as an expedient to reloop through or exit from the program. This process detracts from the educational benefit and timeliness of DMS.

As part of the training simulation, monitors spend many hours in analyzing the progress of the teams and the positions for contract negotiation. Analysis done by each team must also be accomplished by the monitor to assess student

D. MONITOR FUNCTIONS

CNP provides one additional table for monitors. COST+FACTORS, Table III , provides the monitor with the cost for a unit increase in a system test category as well as with the design factors achieved.

```
                          TABLE III
                        COST + FACTORS

  ---------------------------------------------------------

  COSTS OF TESTS PER UNIT (IN THOUSANDS OF DOLLARS)
     QUALIFICATION TESTS
                MOTOR :     32.034
            AIR FRAME :    126.355
             GUIDANCE :    309.906
         FIRE CONTROL :    938.186
             LAUNCHER :    424.555
         FLIGHT TESTS :    316.855
  --------------------------
  DESIGN FACTORS ACHIEVED
         COMPONENT      TABLE   ROW    FACTOR    ACHIEVED
             MOTOR :      1      25     0.825     0.877
         AIR FRAME :      2      15     0.958     0.980
        GUIDANCE A :      5      22     0.666     0.630
        GUIDANCE B :      6      25     0.580     0.0
      FIRE CONTROL :      4      16     0.845     0.802
       LAUNCHER :     3      7     0.936     0.988

   IHO001A PAUSE ;PRESS <<ENTER>> TO CONTINUE.
```

progress. The monitor must also create positions from which to interact with the teams as the contractor. Within the COST+FACTORS Table, some aid exists for the monitor. However, his understanding of the impact of unit-cost on the contract requires additional analysis time over what the teams spend on incentives analysis. In general, because of limited time the monitor has to spend analyzing each team's

proposals compared with the amount of time put in by the team, coupled with the inconvenience in making repeated CNP runs, most monitors produce an actual DMS table printout corresponding to each team's proposed decision in order to provide more detailed and accurate information for use during contract negotiations. It would be better for CNP to provide the information in a more usable format than to produce the lengthy and time consuming main exercise reports of the DMS.

E. SUMMARY

The difficulties encountered in CNP detract from its usefulness. Time spent fighting the program problems are excessive compared to the hours available for productive DMS analysis. In an effort to correct these time-sinks in CNP, PROJMNG FORTRAN and its associated EXEC routines have been created.

III. PROJMNG OPERATION

A. CNP MODERNIZATION

The task in creating PROJMNG FORTRAN as a revision to CNP was first to retain the same functional structure relationship with DMS. This goal has been accomplished by incorporating the CNP subroutines into a larger program. The expansion has occurred by the added Fortran code for the human-factors engineered subroutines, and new sensitivity analysis features. The human-factors designs provide additional user friendliness to PROJMNG. The program calculations continue to center around the original CNP program routines. Many of the achieved value parameters and performance calculations are accomplished using the same Fortran code as used by CNP. CNP is encompassed in PROJMNG.

Other tasks for PROJMNG FORTRAN were to correct as many of the discrepancies in CNP as possible, to provide an enhanced user-friendly training device and to improve the analysis capability available to the monitor.

In modernizing CNP, the criticisms specified in Chapter 2 were highlighted. The following areas were examined.

1. Link
2. Database
3. Database Security
4. Exit
5. Data Change
6. Menu
7. Report Submission
8. Monitor Access
9. Sensitivity Analysis
10. Calculation Correction

B. PROJMNG PACKAGE

This chapter discusses the features of PROJMNG used to support team analysis of the exercise. The monitor capabilities are discussed in Chapter 4.

1. Programs

PROJMNG consists of more files (executable programs) than does CNP. There are four routines utilized in executing PROJMNG FORTRAN. In addition to the PROJMNG FORTRAN file, the package includes LINKPROJ EXEC (Appendix C), PROJMNG EXEC (Appendix D), and the PROJMNG disk's PROFILE EXEC (Appendix E) which are discussed in detail throughout this thesis.

PROJMNG FORTRAN is designed to function alone provided team files exist and have been defined for PROJMNG FORTRAN as files 9 and 10: "FILEDEF 09 DISK DATAFILE TEAMxx" and "FILEDEF 10 DISK DATACODE TEAMxx". It can be used without the exec routines but at a lower level of performance. PROJMNG can be executed through the routine PROJMNG EXEC.

PROJMNG FORTRAN and PROJMNG EXEC are designed to reside on a publicly accessible disk. In this manner multiple teams can use the program simultaneously. Their database files will be on this one disk where they are conveniently accessible for the program and the monitors. PROJMNG is read from this disk by the team's user disk and executed from the team's disk.

PROJMNG both reads and writes onto its data files. The IBM-3033 public, "all", link access is read-only (R/O) and prevents writing onto the disk. The PROJMNG disk is only read capable when linked to for the exercise. However, the disk linking to PROJMNG can be written onto by the program. A team disk linking to PROJMNG acts as the write disk for the program.

The ability for teams to write directly onto the PROJMNG disk would be unsatisfactory. This access to the program would permit two detrimental effects. The program might be either altered or destroyed by overly curious students entering the files, and team files might also be altered or destroyed. Inadvertent alteration of the files is prevented by student access being limited to read only.

2. Data files

There are four files of data tables that were created for CNP. These tables contain the resource parameters to generate achieved values for the program calculations. These data tables remain in files FT17F001, FT18F001, FT19F001, and FT20F001 as in CNP. No contract simulation calculations can be accomplished without them. The program will dump when it requires a table be read and does not find it. PROJMNG will also stop if the tables are not in the required format.

Preformatted database files must be available when the STORE and read routines are called by PROJMNG. Whenever it attempts to write or read these data files and does not find them, the program will dump (stop). Two of the remaining files are these database files. One is the FILE DATAFILE which is preformatted for the DMS data. It serves as a dummy file which can be copied to initiate the team's database, DATAFILE TEAMxx. The other file, FILE DATACODE, serves as a similar dummy file in the creation of the team's security file, DATACODE TEAMxx.

DATACODE stores the team security code, the security code count for attempts to breech the code, and the USERID of the last team which used the file. Its data enables the program to use routines to check the team and validate the team identity through the security code.

If formatted files for the team are not available on the PROJMNG disk or the team disk, the dummy files are copied onto the team disk. PROJMNG EXEC reproduces the two dummy files on the team disk in the names DATAFILE TEAMxx and DATACODE TEAMxx. They are used interactively by PROJMNG during the game sessions. The PROJMNG Instructions (Appendix F) discuss the DATABASE and DATACODE file generation on the team's disk.

When the team finishes its session and exits from PROJMNG FORTRAN, the last copy of the data which was stored by the program remains on the team's disk. As the computer execution finishes the routine PROJMNG EXEC, these two files are transmitted to the PROJMNG disk.

3. LINK

LINKPROJ establishes the link between the team's disk and the PROJMNG disk. It requires PROJMNG EXEC in order to run the program.

The routine LINKPROJ is a CMS executive level program which utilizes the link feature to establish the connection between the team's disk and the disk on which PROJMNG resides. It begins the recording of the team's session, "RECORD ON". This first step in LINKPROJ provides a record of the session for review in the event PROJMNG does not perform as desired. Once the link has been established, the routine sets the function of the programmable function key 9 (PF9) to enable release of the link. This feature enables the team to release the link after a session has been aborted and the program stopped without completing the LINKPROJ EXEC. -The team would have to abort a session should PROJMNG be stuck in a continuous loop.- PF9 will not succeed in breaking the link when PROJMNG is operating normally.

Team members should be aware of the IBM-3033 capability to be dumped (stopped) out of CMS by executing <<ALT>> and <<PA1>> simultaneously. Executing <<PA1>> aborts the program without completing LINKPROJ. The pressing of these two keys on the IBM 3278 terminals will place the virtual system into the CP level of operator interaction. The team can reaccess its "A" disk by executing the command "I CMS" and pressing <<ENTER>>; wait for the display to read "VM READ" and again press <<ENTER>>. The disk will again be in CMS and existing links will remain intact. In order to release the link to PROJMNG press the <<ALT>> and <<PF9>> simultaneously.

LINKPROJ breaks the link. It releases the PROJMNG disk (0276P). Once the disk is released if no users are accessed to it, the PROJMNG disk can be logged onto in the read/write mode to enable files to be written onto it.

When LINKPROJ sends the special message to PROJMNG's disk, "SMSG QACNT KLINE" it tells the computer to automatically log (autolog) onto the disk. The disk is activated in the read/write mode only if no other user is on the PROJMNG disk. The files which the team sent are copied by the disk's PROFILE EXEC.

LINKPROJ also terminates the recording of the session, "RECORD OFF".

4. PROJMNG EXEC

a. Initiate Files

PROJMNG EXEC performs the checks of available files. It determines if a formatted data file and a security file are available for the team. If it detects the monitor unique file, DATAINST DISKNUM, it initiates the monitor flag to tell PROJMNG FORTRAN the user is a monitor.

PROJMNG EXEC checks first for the monitor's file, determines if the user is a team or a monitor. If the user is a team, it checks for the team files on the PROJMNG disk and uses them if available. If these files are not found, it will check the team disk for the files and use them. If the files are not found at all, the program copies the dummy files and generates the team label and files.

b. Terminating the Session

If the user is a monitor, PROJMNG EXEC queries him for whether he desires a concatenated listing of all of the team files. If he does, it spools a file with all team data files on it. It then completes its execution and returns operation to LINKPROJ.

For team operation, when the computer leaves PROJMNG FORTRAN control, execution returns to PROJMNG EXEC. In order to complete the session, PROJMNG EXEC has been designed to transmit the data files to the PROJMNG disk. This feature is accomplished by spooling the disk, "SPOOL PUNCH CONT CL X 0276P". The punch file created is sent to userid 0276P as a class x file. PROJMNG EXEC punches the data files onto the spool, "PUNCH DATAFILE TEAMxx", and closes the spool, "SPOOL PUNCH CLOSE NOCONT". The file is queued in the reader file of the PROJMNG disk (0276P).

5. PROFILE EXEC

When LINKPROJ sends the automatic log-on message to the PROJMNG disk, if no other user is on the PROJMNG disk, the disk is activated in the read/write mode. The main computer executes the disk's profile routine. PROFILE EXEC is performed whenever the disk is accessed in the write capable mode. This exec has been uniquely tailored for this PROJMNG specific disk (see Appendix E). It reads up to six class x files queued to the disk. First the PROFILE EXEC

sets the disk reader file to look for class x, "SPOOL RDR CL X", and then reads the files, "READ *". The program has been designed to reset the reader spool to spool any class. Upon completion of writing files onto the disk, the command "SPOOL RDR CL *" returns normal spooling capability

The PROFILE read function would nominally expect to find a set of files, DATAFILE TEAMxx and DATACODE TEAMxx. The read limit was expanded from two files to six, for three pairs, in expectation that there will be occasions when the disk will remain in use by other teams. When other teams are still using the disk the autolog will be unable to perform its one shot access to the disk. SMSG will not remain queued to access the disk. In order to read those class x files that may have been left waiting, subsequent running of the PROFILE EXEC will read in several more files than those transmitted in the current PROJMNG session. The exec will read up to three additional pairs.

The benefit in using the class x spooling capability is the cue which this class gives. It tells the disk that the file is a PROJMNG file. Normal spooling usage within the IBM 3033 does not use x. 'X' has a uniqueness within the IBM system and will reduce the possibility of PROJMNG EXEC reading non-PROJMNG files.

The files are not read onto the PROJMNG disk while other users remain on the disk. Attempts to write onto the disk in this situation result in garbling the data files.

C. DATABASE SECURITY

The first query the team answers in PROJMNG is to provide the team number.

The second question asks for the team to create a team security code. The team may enter any code of up to eight characters. If the eight characters are all zeros or

contain nothing (nulls) the program will again ask the team to change the code. PROJMNG begins succeeding sessions by asking the team to match the security code stored in data base, the team's security code.

The security code system gives the team access to its database while assuring that data has not been altered by anyone except the team or the monitor. This has benefits to the team and to the simulation. The team doesn't have to worry about data being changed accidently and the monitor has less concern that the team's work might be compromised. The security code is also used for proposal submission. Due to its length and improbability of being entered correctly by mistake, the security code performs the function of assuring the monitor of the team's intent to submit the proposal as indicated.

The monitor can change the database to correct errors in the team files. Corrections may become necessary if the team submits a report and then wants to change it. The proposed and final data reports for a DP are not accessible by the team. Lack of access to submitted reports increases team commitment to the report. It also, provides the monitor with a fixed team position reference point. This is critical if the monitor is to create a contractor position by which to interact with the team.

The submission of reports and their purpose are discussed in greater detail later in this chapter.

D. EXIT

In PROJMNG several methods of stopping the program have been provided. These are:

1. Fatal errors
2. Security violation
3. Menu option

4. Yes/No query

1. Fatal Errors

The computer capability to stop the program when a programmed statement cannot be performed (executed) is referred to as a fatal error. PROJMNG has selectively modified the fatal error response.

The program termination in response to a null entry (pressing the <<ENTER>> key without providing a character including the character blank ' ') was unsatisfactory. Inadvertent null responses repeatedly dumped users out of the program.

To resolve the fatal error dump (stopping the program), a GLOBAL routine (program which can be called from the IBM 3033's resident routine files) is called. The 'CALL ERRSET' enables the program to ignore fatal errors and proceed with the program. When PROJMNG finds an error during the process of reading information from the terminal, it performs a jump to the statement specified as the error default by the READ statement. Very serious errors in syntax will continue to be dumped by the computer.

Failure to answer PROJMNG's questions satisfactorily will cause it to reloop to either the same question or to call out the EXITS routine. If an error is serious enough, the EXITS routine will be called. EXITS allows termination of the program, or if the team wants to continue, EXITS will return to the top of the last query routine and allow the program to reloop. Correct entry of the security code will give the team access to the main program and existing data files with that team's number.

2. Security Violation

During the start of a session at the terminal, the user will be queried for his previously entered security code. The program will permit the user to answer the security code query incorrectly only five times during the course of a DMS exercise. At occurrences of incorrect security code entries, a count is kept. On the fifth occasion when the code is entered incorrectly, the counter will tell the program to stop. Successive attempts to enter a team file whose security file counter has reached five will terminate at the security code query.

3. Menu

A standard option in the main menu of PROJMNG is EXIT. Selecting the EXIT option directs the program to stop. Again, this operation returns computer execution to the PROJMNG EXEC program. Completion of the PROJMNG EXEC run will return control to the LINKPROJ routine which presents the query for another run.

4. YES/NO Query

Each query in PROJMNG which can be answered with a Y for YES or N for NO, can also be answered with an E for EXIT. This E response either calls the program subroutine EXITS or stops the program. EXITS gracefully stops the program by permitting either return of the computer's operation to PROJMNG EXEC or by relooping into the last query routine.

5. Return to PROJMNG EXEC

The termination of PROJMNG FORTRAN returns the computer execution to PROJMNG EXEC. This routine executes a transmittal of all team files to the PROJMNG disk.

E. PROJMNG PROCEDURES

1. Instructions

The procedures used with PROJMNG FORTRAN are significantly different from CNP. Appendix F contains the PROJMNG Student Instructions. These instructions replicate the program's screen displays. Appendix F demonstrates a session using PROJMNG. Checks and corrective procedures in PROJMNG render the making of errors more difficult. However, some examples are demonstrated in the PROJMNG Student Instruction's tutorial.

2. Link

The instructions ask the team to access PROJMNG solely by executing the command LINKPROJ. This command runs the program LINKPROJ EXEC. LINKPROJ EXEC is a routine which creates the disk link and accesses PROJMNG as the user's 'C' disk, as discussed above. This exec must be placed on the team disk. From there it is able to link and access PROJMNG's disk for the team.

Among the capabilities of LINKPROJ are the definition of programmable key PF9. This function can be executed by pressing <<ALT>> and <<PF9>> simultaneously.

LINKPROJ also executes a pair of calls to the macro computer program routine "RECORD". The first of this pair enables the recording of all session terminal activity. The latter executes after the session has been completed; it terminates the record of the session. Teams should be aware that if the program dumps or they exit the program without performing the "RECORD OFF", the record will continue in effect. It can be closed by inputting the command "RECORD OFF". Termination of the record permits the team to either name and save the session's record on disk, to edit it, to print it, to quit it ("Q") which places it on the disk

spool, or to purge the record. Addition of this capability permits review of team session. Undetected difficulties or errors that result in program malfunction can be reviewed.

In order to facilitate rerunning of LINKPROJ, the routine does query whether the team wishes to run "ANOTHER RUN, Y/N". This feature is one which LINKPROJ holds in common with the other programs in the PROJMNG package of routines. PROJMNG is designed to allow the team to terminate or reloop through program execution.

The running of PROJMNG FORTRAN does not occur from LINKPROJ. LINKPROJ runs limited functions already specified, and calls the PROJMNG disk's program PROJMNG EXEC.

3. PROJMNG EXEC Execution

PROJMNG EXEC serves several functions in preparing the team disk for the session.

a. Team Disk Status

The PROJMNG EXEC routine provides several queries to the team disk. Through the response error return codes it receives indication of the availability of team data base files and security code files. The purposes accomplished by this procedure are:

(1). *File Availability*. The error return codes tell PROJMNG EXEC whether there are files available for use. Neither PROJMNG EXEC nor PROJMNG FORTRAN can proceed without files from which data can be read or to which data can be written.

(2). *File Integrity*. Having found a usable DATAFILE on the team disk, PROJMNG EXEC attempts to read the contents. It looks specifically for the team number. The team number cues PROJMNG EXEC not only as to what the team should be called, it also reveals the prior use of the DATAFILE to store data.

b. Monitor File

In a similar manner, PROJMNG EXEC determines the availability and validity of the monitor file "DATAINST DISKNUM". The presence of data in the file and the security field format within DATAINST DISKNUM tells PROJMNG if the disk gaining access to it belongs to a monitor. Disks belonging to monitors are not intended to be the disk which students access to run PROJMNG.

c. DATAFILE Creation

PROJMNG EXEC determines the team disk status. If DATAFILE and DATACODE files are not found, PROJMNG EXEC proceeds to determine the team number. It will ask "WHAT IS YOUR TEAM NUMBER?" before it copies the files. After the team number is evaluated for being in the range of 1 to 20, the program generates the "TEAMxx" label. It will accept 1 to 20 with any number of preceding 0's. The TEAMxx label will include up to four numbers in the xx field. This label is used in the process of copying the dummy files. PROJMNG copies the dummy files using the labels DATAFILE TEAMxx and DATACODE TEAMxx as the new file name and file type.

In the event that more than 20 team files are required, PROJMNG will accept team file labels with additional leading zeroes. Teams with files labeled TEAM01 and TEAM001 can both be created.

The availability of the DATAFILES also gives PROJMNG EXEC the team number. Lack of DATAFILES is synonymous with no team number. The team number is determined only once during the course of the gameplay. It is PROJMNG EXEC that makes that query. It forms the team number into the "xx" portion of the files' type labels. PROJMNG stacks the team number into its virtual reader to enable PROJMNG FORTRAN to copy the team number. PROJMNG FORTRAN uses the

team number to label displays and, when storing the database, places the number at the head of the record. It is this storing process and filling in of the first DATAFILE field which later enables PROJMNG EXEC to find the team number without additional queries.

4. Running PROJMNG

PROJMNG EXEC completes determining if the disk belongs to a team or monitor, gets the team number or sets the monitor's flag of team number 21 (FL21), insures the availability of write files or creates them, and then loads and starts PROJMNG FORTRAN's text file. The team database capability has been set for 20 teams. The data file for monitors was chosen in excess of the team database design size of 20. Having chosen TEAM21 to be the monitor's file label, it was convenient to utilize the label TEAM21 as a flag between PROJMNG EXEC and PROJMNG FORTRAN. It indicates to PROJMNG FORTRAN that the computer operator for the session has been determined by PROJMNG EXEC to be a monitor. Upon termination of PROJMNG FORTRAN, PROJMNG EXEC again takes control. If the operator is a monitor, the EXEC allows the monitor to request the creation of a concatenated file of all current team files on the PROJMNG disk. For team disks, the exec sends a copy of the files DATAFILE and DATACODE to the master disk. This file transmission enables the monitor to determine team progress and to accept the DP decision reports. PROJMNG erases unnecessary residual files such as the LOAD MAP and then returns execution control to LINKPROJ to perform the "ANOTHER RUN, Y/N?" query. The team's response to this question will either terminate the session or reloop the program to the beginning of LINKPROJ.

F. DATABASE CHANGE

After data files have been created, PROJMNG uses the DATAFILEs as a baseline of team information. Once the file is created the program assumes the data is accurate and satisfactory for use in all subsequent sessions. All further Selection Menu displays will display the DMS data items as they are currently found in the team's file DATAFILE TEAMxx.

At the beginning of the team's first session all of their data items must be submitted. A database is generated from which successive sessions will obtain their baseline of information. The process of asking for all data items occurs through the program's subroutine PAGEIN.

The pages of the DMS decision sheets, Appendix E, list the data entries necessary for CNP. These same entries are queried once in PROJMNG by PAGEIN as demonstrated in Appendix F. Each page of data queried coincides with the page and the data item of the decision sheets. Available parameter ranges are listed with the item name in PAGEIN. Correct completion of a page of data permits the program to check the data input for errors. If the program does not understand the input, or requires revision, a *****WARNING***** is displayed and questions designed to correct the discrepancy are asked. If the data entered cannot be read by PROJMNG the EXIT query "DO YOU WISH TO CONTINUE: Y/N?" will occur. Teams experiencing confusion, or wishing to exit the program during PAGEIN, can do so by pressing <<ENTER>> without any input. During PAGEIN this action will enable the team to use the exit query to stop the program.

The automatic call to the PAGEIN routine occurs only when the DATAFILE contains 0's in all of the data base data fields. Successive sessions in PROJMNG by a team utilize

the database DATAFILE to fulfil the data field input function. PAGEIN can be called at any time from the Selection Menu. This option permits the team to make mass changes to data items.

Once the program has a set of data it will produce a table containing those values, Table IV. The SELECTION MENU gives a review of the current data values and acts as a selection menu for items the team wishes to change. SELECTION MENU presentation is achieved by either completion of PAGEIN, by beginning a subsequent session, by completion of a DP proposal submission, or by team choice from the MAIN MENU.

```
DO YOU WISH TO:

1. RECEIVE THE TABLE OF ACHIEVED VALUES.
            *****WARNING*****AFTER THE TENTH DP- 3 RUN
            AN ADDITIONAL DEVELOPMENT COST OF $100,000
            WILL BE INCURRED FOR EACH ADDITIONAL RUN.
            THIS IS RUN  1 OF YOUR DP- 3.
2. INPUT SELECTION MENU.
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT
```

Figure 3.1 MAIN MENU.

G. MENU

Continuing sequentially through an initial terminal session with PROJMNG, the next feature which differentiates this program from CNP is the MAIN MENU, Figure 3.1. From this routine in the main program, any of the options performed by PROJMNG for the team can be executed. This menu places the PROJMNG FORTRAN program in the category of being a menu-driven program.

TABLE IV

DATABASE PRINTOUT as SELECTION MENU

```
NO.*ITEM******CURRENT VALUE @ DP-3  NO.*ITEM***********CURRENT VALUE
 1. CONTRACTOR            :  1      16. GUIDANCE QUAL TESTS  :  6.00
 2. GUIDANCE APPROACH     :  1      17. FLIGHT TESTS         : 17.00
 3. GUIDANCE CONFIGURATION:  2      18. MAXIMUM COST       M$: 53.00
 4. MAINTAINABILITY ENG $M:  1.75   19. MINIMUM COST       M$: 47.00
 5. VALUE ENG           M$:  0.75   20. MAX COST INCENTIVE  %:  4.00
 6. PARALLEL DEVELOP    M$:  0.0    21. LATEST WEEK          :238.00
 7. MOTOR RELIABILITY    %: 92.00   22. EARLIEST WEEK        :202.00
 8. AIRFRAME RELIABILITY %: 97.00   23. MAX DELIVER INCENTIVE%:  3.50
 9. LAUNCHER RELIABILITY %: 98.50   24. MAXIMUM RELIABILITY %: 81.00
10. FIRE CONTROL ERROR YDS: 80.00   25. MINIMUM RELIABILITY %: 75.00
11. GUIDANCE ERROR     YDS: 70.00   26. MAX RELIABIL INCENTIV%:  3.50
12. MOTOR QUAL TESTS      : 25.00   27. MAXIMUM ERROR     YDS:160.00
13. AIRFRAME QUAL TESTS   :  6.00   28. MINIMUM ERROR     YDS:140.00
14. LAUNCHER QUAL TESTS   :  3.00   29. MAX ERROR INCENTIVE %:  4.00
15. FIRE CONTROL QUAL TSTS:  3.00   30. WEEK FOR LOT 10      :337.00
              *****0. NONE**********31. INPUT BY PAGEIN
        SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3
```

The options available in the MAIN MENU are to print the table of achieved values, to return to the database table, to submit a proposal, or to exit.

The achieved values option directs the program to execute the routines associated with creating the Development Contract Summary table, Table V. First the data entries for fee percentages are checked. The fee percentages are added together and evaluated to determine if they total 15%. If they do not equal 15%, PROJMNG FORTRAN requests corrections through a sequence which displays each current fee percentage value and asks for a value to be submitted. The program performs the computations for the table created in subroutine PROSUM. Then the table is displayed. Each time the computations are made the run is counted. After ten runs, the program adds a fee of $100,000 onto the contract price for each subsequent run at that DP level.

The second option returns the display of the data table. It provides the team with the table, Selection Menu, of data items which are used for reference in changing data values.

The third capability from the MAIN MENU is proposal submission. Submission of proposals is discussed in the next section.

To enable the team to stop the program, the fourth option in MAIN MENU allows them to exit.

B. REPORT SUBMISSION

1. Progress Reports

Submission of progress reports has been readily accomplished in PROJMNG EXEC. The process of completing PROJMNG EXEC when leaving PROJMNG FORTRAN serves to execute the commands for file transmission as discussed

TABLE V

Development Contract Summary

```
-----------------------------------------------------------------------------

DP-3 *** DEVELOPMENT CONTRACT SUMMARY *** TEAM  3

                 INCENTIVE PROVISIONS              INCENTIVE ACHIEVEMENTS

    INCENTIVE      WORST     BEST     MAX FEE     ACHIEVED      FEE       FEE %
      AREA         VALUE     VALUE    ALLOWED      VALUE       EARNED     EARNED

    DEV. COST     $ 53.0M  $ 47.0M     4.0%     $ 48.54M    $ 1.486M    2.97%
    FLT TST COMPL  238 WK   202 WK     3.5%       199 WK    $ 1.750M    3.50%
    RELIABILITY    75.0%     81.0%     3.5%      80.08%     $ 1.480M    2.96%
    ACCURACY       160YDS   140YDS     4.0%      143YDS     $ 1.681M    3.36%

    TOTALS                            15.0%                 $  6.397M  12.79%

    TOTAL CONTRACT PRICE            = $ 54.9M

    IH0001A PAUSE ;PRESS <<ENTER>> TO CONTINUE.
```

in subsection 3.B.4.2. The process of data file transmission to the PROJMNG disk provides those files for access by the monitor. In the monitor's session, all team data files which have been sent to the PROJMNG disk are copied onto the monitor's personal disk for read/write use. This feature was also discussed in 3.B.5. and is covered in greater detail in Chapter 4 with the discussion of monitor functions.

2. Decision Submission

Submission of the team's decision occurs with the transmittal of the team files to the PROJMNG disk. Within the data fields transmitted, there is a label field which contains the status of the team's work on a given decision point. This field is either blank, "00000000", "PROPOSED" or "FINAL". These labels are sequentially exclusive. They specify the exact status for the DP level of the set of database fields in which they are contained. Each DP level has four lines of data in the DATAFILE TEAMxx. The DP status label is the last field of data in a set of lines.

The DP field labels are assigned in the FINISH subroutine. By accepting the values in the table of proposed values for the contract proposal, Proposed Values Table (Table VI), or of proposed final values as the final approved contract, Final Contract Values Table (Table VII), the team causes the label to be written. These labels occur only when the team enters its security code to validate the proposed values. Completion of the value assignments to the database for transmittal appears with the table of accepted values, Table VIII .

The label is written as the last field in each of DP level database records. These labels can be seen in the 33rd field of each set of four lines of a DP data, Table IX.

TABLE VI

Proposed Values Table

```
NO.*ITEM******CURRENT VALUE @ DP-3  NO.*ITEM************CURRENT VALUE
 1. CONTRACTOR            :  1      16. GUIDANCE QUAL TESTS    :  6.00
 2. GUIDANCE APPROACH     :  1      17. FLIGHT TESTS           : 17.00
 3. GUIDANCE CONFIGURATION:  2      18. MAXIMUM COST         M$: 53.00
 4. MAINTAINABILITY ENG $M:  1.75   19. MINIMUM COST         M$: 47.00
 5. VALUE ENG           M$:  0.75   20. MAX COST INCENTIVE    %:  4.00
 6. PARALLEL DEVELOP    M$:  0.0    21. LATEST WEEK            :238.00
 7. MOTOR RELIABILITY    %: 92.00   22. EARLIEST WEEK          :202.00
 8. AIRFRAME RELIABILITY %: 97.00   23. MAX DELIVER INCENTIVE%:  3.50
 9. LAUNCHER RELIABILITY %: 98.50   24. MAXIMUM RELIABILITY   %: 81.00
10. FIRE CONTROL ERROR YDS: 80.00   25. MINIMUM RELIABILITY   %: 75.00
11. GUIDANCE ERROR     YDS: 70.00   26. MAX RELIABIL INCENTIV%:  3.50
12. MOTOR QUAL TESTS      : 25.00   27. MAXIMUM ERROR       YDS:160.00
13. AIRFRAME QUAL TESTS   :  6.00   28. MINIMUM ERROR       YDS:140.00
14. LAUNCHER QUAL TESTS   :  3.00   29. MAX ERROR INCENTIVE   %:  4.00
15. FIRE CONTROL QUAL TSTS:  3.00   30. WEEK FOR LOT 10        :337.00
***DO YOU WISH TO ENTER THIS DATA AS YOUR PROPOSED DP- 3 DECISION?***
             *****CAUTION*****CAUTION*****CAUTION*****
              YOU MAY ONLY INPUT A DECISION ONCE.
ENTER YOUR TEAM SECURITY CODE IF YOU WISH TO COMMIT TO A DECISION
;OR "CONT".
```

TABLE VII

Final Contract Values Table

```
NO.*ITEM******CURRENT VALUE @ DP-3  NO.*ITEM*************CURRENT VALUE
 1. CONTRACTOR            :  1       16. GUIDANCE QUAL TESTS   :  6.00
 2. GUIDANCE APPROACH     :  1       17. FLIGHT TESTS          : 17.00
 3. GUIDANCE CONFIGURATION:  2       18. MAXIMUM COST        M$: 53.00
 4. MAINTAINABILITY ENG $M:  1.75    19. MINIMUM COST        M$: 47.00
 5. VALUE ENG           M$:  0.75    20. MAX COST INCENTIVE   %:  4.00
 6. PARALLEL DEVELOP    M$:  0.0     21. LATEST WEEK           :238.00
 7. MOTOR RELIABILITY    %: 92.00    22. EARLIEST WEEK         :202.00
 8. AIRFRAME RELIABILITY %: 97.00    23. MAX DELIVER INCENTIVE%:  3.50
 9. LAUNCHER RELIABILITY %: 98.50    24. MAXIMUM RELIABILITY  %: 81.00
10. FIRE CONTROL ERROR YDS: 80.00    25. MINIMUM RELIABILITY  %: 75.00
11. GUIDANCE ERROR     YDS: 70.00    26. MAX RELIABIL  INCENTIV%:  3.50
12. MOTOR QUAL TESTS      : 25.00    27. MAXIMUM ERROR       YDS:160.00
13. AIRFRAME QUAL TESTS   :  6.00    28. MINIMUM ERROR       YDS:140.00
14. LAUNCHER QUAL TESTS   :  3.00    29. MAX ERROR INCENTIVE  %:  4.00
15. FIRE CONTROL QUAL TSTS:  3.00    30. WEEK FOR LOT 10       :337.00
*****************************IMPORTANT********************************
***YOU MUST CHECK THE ABOVE ENTRIES WITH THE APPROVED FINAL DP-3 .***
ENTER YOUR TEAM SECURITY CODE IF YOU WISH TO COMMIT TO A DECISION;
                            OR "CONT".
```

TABLE VIII

Accepted Values Table

```
       ******THE FOLLOWING PARAMETERS WERE ACCEPTED******
       AS TEAM  3 DP- 3 PROPOSED INPUT TO THE CONTRACTOR.
NO.*ITEM******CURRENT VALUE @ DP-3  NO.*ITEM***********CURRENT VALUE
 1. CONTRACTOR             :  1      16. GUIDANCE QUAL TESTS    :  6.00
 2. GUIDANCE APPROACH      :  1      17. FLIGHT TESTS           : 17.00
 3. GUIDANCE CONFIGURATION:  2       18. MAXIMUM COST         M$: 53.00
 4. MAINTAINAEILITY ENG $M:  1.75    19. MINIMUM COST         M$: 47.00
 5. VALUE ENG           M$:  0.75    20. MAX COST INCENTIVE    %:  4.00
 6. PARAILEL DEVELOP    M$:  0.0     21. LATEST WEEK            :238.00
 7. MOTOR REIIABILITY    %: 92.00    22. EARLIEST WEEK          :202.00
 8. AIRFRAME RELIABILITY %: 97.00    23. MAX DELIVER INCENTIVE%:  3.50
 9. LAUNCHER RELIABILITY %: 98.50    24. MAXIMUM RELIABILITY   %: 81.00
10. FIRE CONTRCL ERROR YDS: 80.00    25. MINIMUM RELIABILITY   %: 75.00
11. GUIDANCE ERROR     YDS: 70.00    26. MAX RELIABIL INCENTIV%:  3.50
12. MOTOR QUAL TESTS      : 25.00    27. MAXIMUM ERROR       YDS:160.00
13. AIRFRAME QUAL TESTS   :  6.00    28. MINIMUM ERROR       YDS:140.00
14. LAUNCHER QUAL TESTS   :  3.00    29. MAX ERROR INCENTIVE   %:  4.00
15. FIRE CONTROL QUAL TSTS:  3.00    30. WEEK FOR LOT 10        :337.00
       *****SEE YOUR MONITOR IF YOU HAVE A PROELEM.******

IHO001A PAUSE ;PRESS <<ENTER>> TO CONTINUE.
```

TABLE IX

DATAFILE TEAM03

```
---------------------------------------------------------------------------
3  3 112      1.75    0.75    0.0     92.00   97.00   98.50   80.00   70.00
    25.00     6.00    3.00    3.00     6.00   17.00   53.00   47.00    4.00
   238.00   202.00    3.50   81.00    75.00    3.50
   160.00   140.00    4.00  337.00     6.00   FINAL
3  4 112      1.75    0.75    0.0     92.00   97.00   98.50   80.00   70.00
    25.00     6.00    3.00    3.00     6.00   17.00   53.00   47.00    4.00
   238.00   202.00    3.50   81.00    75.00    3.50
   160.00   140.00    4.00  337.00     6.00   PROPOSED
0  0 000      0.0     0.0     0.0      0.0     0.0    0.0     0.0     0.0
      0.0     0.0     0.0     0.0      0.0     0.0    0.0     0.0     0.0
      0.0     0.0     0.0     0.0      0.0     0.0
      0.0     0.0     0.0     0.0      0.0    00000000
1981P
```

Use of the security code in this feature was designed for 'human factor' engineering. The conscious process of entering the team's security code indicates the user's deliberate entry of the data. Inadvertent acceptance of the database for the team's position cannot occur.

The team does retain the responsibility for reading and understanding the data displayed in the table of data, Table VII.

The process of the team submitting a final contract proposal provides the program with the position agreed to during contract negotiations. Submission of the final report indicates to the program that the team has completed a DP level. PROJMNG assumes the team is ready to proceed to the next DP level, for example DP-4. It utilizes the previous DP level data entries from the submitted final report and prepares the new DP data table. The team is sequenced to the next higher level whenever a final proposal is detected for a given DP level.

I. CALCULATION CORRECTION

As discussed in Chapter 2 several of the computations which were designed for CNP to accomplish were not satisfactory. Consequently their display was suppressed.

In PROJMNG these calculations have been reformulated. Their algorithm is based on the linear extrapolation of a fee over the range of its alloted parameter values. The contract cost for the fee incentive is calculated by:

1. Assume achieving optimum performance results in total fee payment.
2. Assume achieving the worst performance expected results in no fee payment.

3. Determine the ratio of satisfactory achievement; the difference between the achieved value and the worst value expected divided by the range of value between worst and optimum values expected.
4. Multiply the ratio determined by the maximum fee percentage offered for the incentive.
5. Multiply the resulting percentage by the target development cost, the average value of the maximum and minimum desired development cost.

The following equation is used in PROJMNG to determine fee ratio.

FEERATIO=FEEMAX*(abs(WORST-ACHIEVED))/(RANGE) (eqn 3.1)

This formulation for fee costs has proven to be more satisfactory than that of CNP.

The difficulty in CNP arises from numerical manipulation. Parameters in CNP frequently have their power-of-ten changed to permit calculations at different resource multiples. This is particularly true in the case of achieved development cost. Frequent multiplication by 1,000,000 with division by 1,000,000 to return the dollar values to millions of dollars incurs computer roundoff. The roundoff errors when repeated several times become sufficiently large to be apparent in the ratio calculations.

PROJMNG does not use the numbers which the computer used in computing achieved value resources to determine the fee values. Its numbers are taken directly from the database array, and are only mathematically manipulated by this one calculation. This process results in only one iteration of roundoff error.

PROJMNG is not able to eliminate computer roundoff error, but the roundoff effect has been significantly reduced.

J. SUMMARY

PROJMNG's design has its foundation in CNP's routines which were designed around DMS. It uses this foundation from CNP, and develops an extended program. The program extension uses the current IBM-3033 facilities to drive user-directed options in a manner which provides the team control over program execution for completion of the DMS exercise. These features for team use of PROJMNG make PROJMNG easier to use in analyzing team positions. Chapter 4 discusses the analysis capabilities of PROJMNG which are available to the monitor.

IV. PROJMNG MONITOR OPERATION

A. CNP MONITOR UNIQUE FUNCTIONS

CNP has one routine capable of providing support to the monitor for his evaluations of the scenario progress. This singular routine is not available to the teams. The monitor can access the COST + FACTORS Table, Table X as discussed in the previous chapter. Access to the monitor routine occurs by the computer operator's response to the query for "COST + FACTORS ?". This response is coded to provide security of the COST + FACTORS table from team access. In order to preserve its integrity, the security code will not be given or demonstrated here. Correct security code responses to both the first "COST + FACTORS ?" and a second query of "COST + FACTORS ?" are necessary. The first response is a three character code word. The second query response is based on a data field used in the computations which has already been entered for the session. It constitutes a variable security code.

In CNP, an additional monitor capability is provided through the confirmation query, "CONFIRM ?", at the DP number verification sequence, "IS THIS DP NUMBER ** xx;"Y"=YES OR "N"=NO ?" . Response to this query with a fixed single letter, places the operator in the "INSTRUCTOR" mode. Analysis of the program's Fortran code indicates that the INSTRUCTOR mode simply suppresses the queries for confirmation and the repeat back display of input data. Answering the statements "IS THIS DP NUMBER ** xx;"Y"=YES OR "N"=NO ?" with a "I" reduces the displayed queries. It eliminates the repeated displays of "RECEIVED ......: xxxxx; CONFIRM ?". It also eliminates the repeat back printout of input page data.

```
                          TABLE X
                       COST + FACTORS

 ----------------------------------------------------------

COSTS OF TESTS PER UNIT, THOUSANDS OF DOLLARS
   QUALIFICATION TESTS
             MOTOR:
  32.034
         AIR FRAME:
 126.355
          GUIDANCE:
 309.906
     FIRE CONTROL:
 938.186
          LAUNCHER:
 424.555
     FLIGHT TESTS:
 316.855

 ------------------------

DESIGN FACTORS ACHIEVED
      COMPONENT     TABLE  ROW  FACTOR  ACHIEVED
             MOTOR:
   1     25   0.825
  0.877
         AIR FRAME:
   2     15   0.958
  0.980
        GUIDANCE A:
   5     22   0.666
  0.630

        GUIDANCE B:
   6     25   0.580

      FIRE CONTROL:
   4     16   0.845
  0.802
          LAUNCHER:
   3      7   0.963
  0.988
```

The loss of page data repeat back and confirmation, and the format of Table X have not been appealing to users of the program. The "T" letter code system was not human factor engineered. Loss of the data item confirmation sequence increases the occasions when inaccurate results are received in the Development Contract Summary Table. Users

have not found benefit in these options. Additional experience with CNP is necessary to use these features successfully. Consequently, use of the "T" feature has not been used. The COST + FACTORS table has been found to be less than adequate.

B. PROJMNG MONITOR UNIQUE FUNCTIONS

In PROJMNG a variety of options are available for the monitor. They include:

1. Monitor Unique Security System
2. Query Suppression for the Monitor
3. Monitor's MAIN MENU
4. File Access Options
5. COST + FACTORS
6. Sensitivity Analysis
7. Graphing
8. Copies of Student Files

This chapter will review the intended capability and operation of these monitor unique functions.

C. MONITOR UNIQUE SECURITY SYSTEM

The PROJMNG EXEC routine looks through the computer operator's disk for a file titled "DATAINST DISKNUM". Having located this file, the computer assumes that the operator is a monitor, and proceeds to validate the monitor identity. The program contains three file checks to determine if the operator is a monitor. Unsuccessful completion of the first two checks will cause the computer to ignore the DATAINST DISKNUM file and assume the computer user is a student.

The first of these checks is the file format. The file first line must contain the userid. The second line must contain a monitor specified security code. This monitor security code is queried in the same manner in which the team security code is queried for a team disk.

The third verification is a check of the stored monitor security code against the individual's ability to reproduce that code. A code match allows the program to proceed. Failure to accurately match the code causes the program to stop. PROJMNG EXEC takes control to finish its execution. The user can reaccess PROJMNG FORTRAN in his response to "ANOTHER RUN, 'Y'/'N' ?".

D. QUERY SUPPRESSION FOR THE MONITOR

Successfully accessing the program as a monitor suppresses the student's Main Menu, discussed in chapter 3. Consequently, the monitor does not receive the student's Main Menu after each alteration of the Selection Menu. Based on the assumption that the computer operator is a monitor the program immediately follows the Selection Menu with the Development Contract Summary, Table XI.

Table XI demonstrates the display sequence for the Selection Menu with the execution of option "0". The monitor's response to the Selection Menu with a zero causes the immediate execution of the achieved value calculations. Other selections from this table are performed similarly to those done for the team. The changes entered by the monitor in this first mode will not be stored in the database by the program. In monitor mode 1, the call to the routine STORE does not occur. Only the array of data in the disk's buffer will be altered. The monitor sees the computer response to his entry of a "0" as an immediate display of the Development Contract Summary.

TABLE XI

SELECTION MENU

```
NO.*ITEM*****CURRENT VALUE @ DP-3  NO.*ITEM************CURRENT VALUE
 1. CONTRACTCR             :  1     16. GUIDANCE QUAL TESTS    :  6.00
 2. GUIDANCE APPROACH      :  1     17. FLIGHT TESTS           : 17.00
 3. GUIDANCE CONFIGURATION:  2      18. MAXIMUM COST        M$: 53.00
 4. MAINTAINABILITY ENG M$:  1.75   19. MINIMUM COST        M$: 47.00
 5. VALUE ENG           M$:  0.75   20. MAX CCST INCENTIVE   %:  4.00
 6. PARALLEL DEVELOP    M$:  0.0    21. LATEST WEEK            :238.00
 7. MOTOR RELIABILITY    %: 92.00   22. EARLIEST WEEK          :202.00
 8. AIRFRAME RELIABILITY %: 97.00   23. MAX DELIVER INCENTIVE%:  3.50
 9. LAUNCHER RELIABILITY %: 98.50   24. MAXIMUM RELIABILITY  %: 81.00
10. FIRE CONTROL ERROR YDS: 80.00   25. MINIMUM RELIABILITY  %: 75.00
11. GUIDANCE ERROR     YDS: 70.00   26. MAX RELIABIL INCENTIV%:  3.50
12. MOTOR QUAL TESTS      : 25.00   27. MAXIMUM ERROR      YDS:160.00
13. AIRFRAME QUAL TESTS   :  6.00   28. MINIMUM ERROR      YDS:140.00
14. LAUNCHER QUAL TESTS   :  3.00   29. MAX ERROR INCENTIVE  %:  4.00
15. FIRE CONTROL QUAL TSTS:  3.00   30. WEEK FCR LCT 10        :337.00
          *****0. NONE**********31. CHANGE BY PAGE*****
       SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3.
?
0                                 <==== "NOTE: The query '?' response "0".

DP-3 *** DEVELOPMENT CONTRACT SUMMARY *** TEAM  3

               INCENTIVE PROVISIONS          INCENTIVE ACHIEVEMENTS

 INCENTIVE     WORST      BEST      MAX FEE   ACHIEVED     FEE       FEE %
   AREA        VALUE      VALUE     ALLOWED   VALUE       EARNED    EARNED

DEV. COST     $ 53.0M   $ 47.0M      4.0%    $ 48.54M   $ 1.486M    2.97%
FLT TST COMPI  238 WK    202 WK      3.5%     199 WK    $ 1.750M    3.50%
RELIABILITY     75.0%     81.0%      3.5%     80.08%    $ 1.480M    2.96%
ACCURACY       160YDS    140YDS      4.0%     143YDS    $ 1.681M    3.36%

TOTALS                              15.0%               $ 6.397M   12.79%

TOTAL CONTRACT PRICE              = $ 54.9M
```

E. MONITOR'S MAIN MENU

The PROJMNG monitor's MAIN MENU contains options to allow the monitor access to any of the routines in the program. It has a unique feature in its flexible length. This feature provides menu listings to the monitor which are determined by his status in the program. If the monitor has just entered PROJMNG or has changed the team number, and has not completed the computations for other display values, the program will suppress those options from the menu display. Options which require calculations or selections which have not been processed for the team are assumed to present results which are either not applicable or misleading.

```
MAIN MENU:
 1. CHANGE TEAM NUMBER
 2. THE DP SELECTION QUERY
 3. THE INPUT SELECTION MENU
 4. RERUN THE DATA CALCULATION
 5. DO A SENSITIVITY ANALYSIS
 6. PRINTOUT COST+FACTORS
 7. EXIT
```

Figure 4.1 Monitor's Main Menu.

Figure 4.1, demonstrates the basic monitor's Main Menu.

1. Change Team Number

Option 1 when selected will return the monitor to the program's team input routine (TEAMIN). The execution of this routine begins by displaying a menu, Figure 4.2. The selections on this menu are:

```
SELECT WHICH INSTRUCTOR MODE YOU DESIRE.
1. RUN TEAM SCENARIOS WHICH ARE ON THE MONITOR'S DISK.
2. CHANGE TEAM FILES.
        THIS MODE WILL ALTER THE STUDENT'S FILES.
3. RUN THE MONITOR'S TEAM FILE 'TEAM21'.
4. EXIT
```

Figure 4.2 Monitor Mode Menu.

a. Run Team Scenarios from the PROJMNG Disk

Selecting this choice enables the monitor to run evaluations and displays of the teams whose DATAFILEs have been written on the PROJMNG disk. PROJMNG has at this point already written these files to the individual monitor's disk for use. As the monitor selects a team to review, the program determines if the team file exists. If the file exists, the monitor may proceed with the evaluation. If no files exist for the team requested, PROJMNG responds with 'FILES DO NOT EXIST FOR TEAMxx', and proceeds to ask for a different team number, 'WHAT TEAM NUMBER ?', until a team number is submitted for which files are present.

This option selects a mode of operation in which team data files are not changed by the monitor. The menus displayed are only those associated with the monitor's options. The Monitor's Main Menus are flexible menus. The discussion of this flexibility and of the basic menu will be provided in the following sections of this chapter. These differences are demonstrated by the two monitor menu displays for mode 1, Figure 4.3, and for mode 2, Figure 4.4

```
MAIN MENU:
 1. CHANGE TEAM NUMBER
 2. THE DP SELECTICN QUERY
 3. THE INPUT SELECTION MENU
 4. RERUN THE DATA CALCULATION
 5. DO A SENSITIVITY ANALYSIS
 6. PRINICUT COST+FACTORS
 7. EXIT
```

Figure 4.3 MAIN MENU in Mode 1.

```
MAIN MENU:
 1. CHANGE TEAM NUMBER
 2. THE DP SELECTION QUERY
 3. THE INPUT SELECTION MENU
 4. RERUN THE DATA CALCULATION
 5. DO A SENSITIVITY ANALYSIS
 6. PRINICUT COST+FACTORS
 7. FILE TEAM PROPCSAL
 8. EXIT
```

Figure 4.4 MAIN MENU in Mode 2.

b. Change Team Files

Option 2 has an entirely different affect on the database. Again, the files accessed are those previously in existence. The monitor receives a different version of the MAIN MENU, The added option in this menu, "7. FILE TEAM PROPOSAL" permits the monitor to change the DP report level for any of the team's DPs irregardless of the team's current level.

The monitor can perform all of the functions he could in mode 1 plus changing the report level and placing these selections in the team database. This mode should only

be used to correct team errors. Team database manipulation for any other reason will lead to team confusion.

The query which finalizes the replacement of the team database, "SHOULD THE REWRITTEN TEAM xx FILE BE SENT TO THE PROJMNG disk: Y/N ?". Unless this query is answered in the affirmative, the mode 2 capability to overwrite the team file is not performed.

Answering yes to the query to rewrite the team file directs PROJMNG through a routine which performs three calls to the subroutine FRTCMS which permits CMS executive level commands to be executed from within a Fortran program. The three commands in this routine are the same as the spool commands in PROJMNG EXEC which send the file DATAFILE TEAMxx to the PROJMNG disk. The disk punch file is spooled as a class x file. The DATAFILE TEAMxx is punched into the spool. Finally the spool is closed and sent to the PROJMNG disk.

The program does allow the DATAFILEs on the monitor's personal disk to be rewritten whenever mode 2 operation is selected. Transmission of these files to the master PROJMNG disk can be accomplished after responding in the negative to the query "SHOULD THE REWRITTEN TEAM xx FILE BE SENT TO THE PROJMNG disk: Y/N ?". Sending these files to the master disk must be a deliberate act of the monitor. The monitor may either use one of the macro routines in general usage on the IBM-3033 for sending files between users, "SEND". Or, he may spool the files for immediate read by the PROJMNG file using the following sequence of steps. (The lower case characters in the following command demonstration are optional inputs.)

1. "SPool PUNch CONT CLass X 0276P"
2. "PUNch DATAFILE TEAM(XX)", where xx is the two digit team number.
3. "SPool PUNch CLOSE NOCONT"

In mode 1 or mode 2 operation, only the Development Contract Summary table resembles the same display for the teams.

c. Run the monitor's team file "TEAM21"

The third option provides a team file for the monitor. TEAM21 is the monitor's own team file to simulate the team interaction with the program. This provides a scratch pad file for the monitor's use.

d. EXIT

Option 4 from the monitor's Main Menu stops the program. Entering a '4' in response to this menu directs the program to stop. PROJMGN EXEC resumes control of the computer processing.

2. The DP Selection Query

Number 2 on the monitor's mode menu reloops the program to the DP query for which DP the monitor desires to evaluate. This query will not prevent the monitor from accessing DP levels for which the teams have not submitted files. The level which the team has completed is displayed on the Selection Menu, with the DP query, and on the proposed report table. This information should be known by the monitor based on the progress of the DMS scenario.

3. The Input Selection Menu

Again, the menu routine simply reloops the program execution to a previously executed question to permit the monitor to change his mode of analysis. In this case he returns to the team Selection Menu. The last team and DP inputs entered during the monitor's session will apply to the team database which is displayed.

Changes to a team's list of data in the program buffer which have been entered during a session remain applicable until the team number is changed. Any variations to the Selection Menu which have been selected for the team currently being evaluated will be in effect. In mode one, a team's data will revert to the PROJMNG disk's team datafile values when changing team numbers. Mode 2 will keep the changed values until the PROJMNG disk values are recopied down onto the monitor's disk.

4. Rerun the Data Calculation

This option causes PROJMNG to redo the achieved value calculations. The display for this selection is a repeat of the Development Cost Summary.

5. Do a Sensitivity Analysis

This option creates a major option for exploration by the monitor. It is discussed as part of the sequential demonstration of the monitor functions which follows in Section I below.

6. Printout Cost + Factors

Similar to the capability for COST + FACTORS demonstrated by CNP, PROJMNG has an equivalent feature for the monitor. Table XII shows the improved layout of the data. The table has the benefit over CNP's version by not requiring a separate code to access it. The table is simply accessed as a menu selection from the monitor's Main Menu.

7. EXIT

This is another example of PROJMNG's ability to stop the program.

```
                         TABLE XII
                       COST + FACTORS

 ---------------------------------------------------------
 COSTS OF TESTS PER UNIT (IN THOUSANDS OF DOLLARS)
    QUALIFICATION TESTS
             MOTOR :    32.034
         AIR FRAME :   126.355
          GUIDANCE :   309.906
      FIRE CONTROL :   938.186
          LAUNCHER :   424.555
      FLIGHT TESTS :   316.855
 ----------------------------

 DESIGN FACTORS ACHIEVED
        COMPONENT     TABLE   ROW    FACTOR    ACHIEVED
            MOTOR :     1      25     0.825     0.877
        AIR FRAME :     2      15     0.958     0.980
       GUIDANCE A :     5      22     0.666     0.630
       GUIDANCE B :     6      25     0.580     0.0
     FIRE CONTROL :     4      16     0.845     0.802
         LAUNCHER :     3       7     0.936     0.988

 IHC001A PAUSE ;PRESS <<ENTER>> TO CONTINUE.
```

8. Flexible Menu

The options presented in this section are the basic capabilities of PROJMNG. However, additional capabilities become possible as the evaluation progresses. With the completion of the sensitivity analysis additional tables and graphs are within the capability of PROJMNG. These flexible menu capabilities are explained and demonstrated in Section I of this chapter.

F. MONITOR SESSION PROCEDURES

The remainder of the discussion on menus will be presented sequentially in the following sections. Their relative position in the textual sequence corresponds to their occurrence in the process of stepping through the program's routines.

These sequential steps of PROJMNG are demonstrated as the program would be operated. The material in this demonstration is preliminary to the preparation of an instructor's manual. The sections are:

1. Initialization
2. Basic routines of the Main Menu
3. Sensitivity analysis
4. Logoff unique routines

G. INITIALIZATION

1. PROJMNG EXEC Operation

The monitor's access to the PROJMNG package of programs is also accomplished by LINKPROJ. The determination that the disk accessing PROJMNG is a monitor is made, as discussed in Chapter 3. Having completed the checks of the monitor's file DATAINST DISKNUM, the PROJMNG EXEC program conducts a check of files for a DATAFILE TEAM21. This file must be present on the monitor's disk if he is going to use it in PROJMNG. PROJMNG EXEC either finds the file or creates one in order to have the file ready for PROJMNG FORTRAN's read and write instructions.

2. Available Files

The lack of presence of the files which PROJMNG EXEC looks for is an automatic function of the IBM-3033. This capability was retained to display to the monitor a list of teams from which he has not received reports, as shown in Figure 4.5 Offsetting of the display from the left margin was performed in order to highlight between the computer printout and the typed queries.

A benefit of the "PRESS <<ENTER>>" set of statements is the ability to begin a display on a fresh screen. Displays which should appear as one screen occur more readily when the pause has finished the previous display to keep it displayed while the user is reading it. The pause can be followed by an immediate command in FRTCMS to 'CLRSCRN ', clear the screen. This sequence of commands leaves a fresh screen for the next screen display. The first line that PROJMNG prints out appears as the top line of a display (see Figure 4.7).

Cueing the user to press <<ENTER>> identifies a pause which has been placed in order to organize screen displays. The pause in Figure 4.5 enables the display of the teams 'not found' to be isolated on one display. There can be up to 21 teams displayed, as for example at the start of an exercise scenario. Without the pause to segregate the displays, all 21 teams would not appear on one screen. The single screen full of one type of data aids in user ability to study and analyze the data. Splitting the display to a second view eliminates the first screen's data and taxes the user's recall to do any comparisons. Isolating the data onto one screen eliminates these disruptions to the user's study of the data. This technique of segregating related information onto a screen has been used throughout this program.

3. Purging DATAFILEs

In order to provide a convenient method of purging files from the completed exercises, a routine was generated which permits the monitor to purge all of the existing files, to conduct normal operation of the program or to exit. These options are depicted in the display as shown in Figure 4.6. The purge option purges only the available 'A' disk DATAFILE TEAMxx and DATACODE TEAMxx. Should the

```
BEGIN RECORDING OF TERMINAL SESSION
* NOTE!: YOU ARE NOW LINKED TO PROJMNG
                              ON YOUR 192 DISK, MODE C
* NOTE!: PRESS PF09 TO BREAK THE LINK
       PRESS <<ENTER>> WHEN YOU ARE READY TO CONTINUE.

------------------------------------------------------

******.....EXECUTION IS IN PROGRESS.
                       WAIT. DO NOT PRESS ENTER.******
FILE 'DATAFILE TEAM21 A' NOT FOUND.
FILE 'DATAFILE TEAM11 C' NOT FOUND.
FILE 'DATAFILE TEAM12 C' NOT FOUND.
FILE 'DATAFILE TEAM13 C' NOT FOUND.
FILE 'DATAFILE TEAM14 C' NOT FOUND.
FILE 'DATAFILE TEAM15 C' NOT FOUND.
FILE 'DATAFILE TEAM16 C' NOT FOUND.
FILE 'DATAFILE TEAM17 C' NOT FOUND.
FILE 'DATAFILE TEAM18 C' NOT FOUND.
FILE 'DATAFILE TEAM19 C' NOT FOUND.
FILE 'DATAFILE TEAM20 C' NOT FOUND.
```

Figure 4.5 Initial Monitor Display.

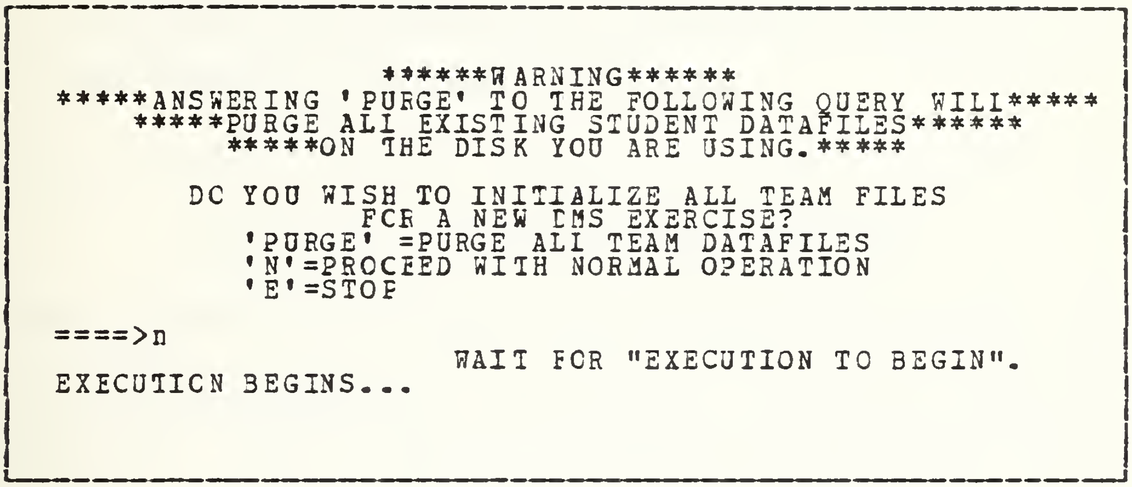
PROJMNG EXEC purge function be performed on the PROJMNG FORTRAN's disk the entire exercise scenario would be eliminated. Only database files pre-existing on team files or monitor's files could be reaccessed for PROJMNG. With a new play of the exercise, it is expected that a new group of teams will be accessing PROJMNG. No conflict with former team databases should be experienced once the monitor's and the PROJMNG's disks have been purged.

Normal operation initiates PROJMNG FORTRAN execution. The standard procedure during a game exercise is designed to respond to this query with an "N". The word length difference between PURGE and N has been designed to make inadvertent destruction of the team files more difficult.

"E" again provides a means of stopping the program. In this instance, "E" will stop PROJMNG EXEC and return the computer to running the LINKPROJ program. The final display

of LINKPROJ, "ANOTHER RUN, 'Y' OR 'N'?", would be the query following the request to stop.

"WAIT FOR EXECUTION TO BEGIN" became a necessary printout as the result of long waits for execution to begin on the NPS IBM-3033. (One monitor reported experiencing a 20 minute delay during very heavy use of the computer facility, with the "WAIT FOR EXECUTION..." label reminding him that the program was in the queue to be run.)



```
                    ******WARNING******
 *****ANSWERING 'PURGE' TO THE FOLLOWING QUERY WILL*****
     *****PURGE ALL EXISTING STUDENT DATAFILES******
          *****ON THE DISK YOU ARE USING.*****

        DO YOU WISH TO INITIALIZE ALL TEAM FILES
                 FOR A NEW DMS EXERCISE?
           'PURGE' =PURGE ALL TEAM DATAFILES
           'N'=PROCEED WITH NORMAL OPERATION
           'E'=STOP

 ====>n
                      WAIT FOR "EXECUTION TO BEGIN".
 EXECUTION BEGINS...
```

Figure 4.6 Monitor Session File Purge Option.

These initial queries, printouts and displays by PROJMNG in the monitor mode of operation should appear to the reader as very similar to those presented to the teams. It should also be noticed in Figure 4.7 that the program does not ask for a team security code input to create the team's code. Instead, it has already received the monitor's security code from PROJMNG EXEC. The monitor's code is found by PROJMNG FORTRAN in DATAINST DISKNUM.

H. BASIC PROGRAM ROUTINES

The program routines discussed in this section perform similar processes in both CNP and PROJMNG.

1. Start of PROJMNG's Run

The stacked data items are read by PROJMNG as its first step in the routine for team number input "TEAMIN".

```
*************************************************************
      TO TERMINATE THE PROGRAM AT ANY POINT,
  TYPE  "E"=EXIT IN RESPONSE TO ANY "YES/NO" QUERY.
*************************************************************
***CAUTION***IN RESPONSE TO A QUERY, PRESSING <<ENTER>>
    WITHOUT PROVIDING DATA WILL DUMP THE PROGRAM.
*************************************************************
?
PLEASE ENTER YOUR SECURITY CODE.
```

Figure 4.7 PROJMNG Initial Displays.

Having received the FL21 flag of TEAM21, and validated the monitor's security code, PROJMNG can proceed. IF the security code should be entered incorrectly, the PROJMNG execution is terminated. The monitor receives a warning of the error, and is asked to reinitiate his startup of the program. He may either answer 'Y' to "ANOTHER RUN, 'Y' OR 'N' ?" or reenter "LINKPROJ" after PROJMNG EXEC and LINKPROJ EXEC have finished execution.

PROJMNG FORTRAN asks for the team number when a monitor is running the program. Team selection within the program is necessary in order to permit the evaluation of more than one team. Whereas a team will only enter the program with one team scenario to perform, the monitor is able to change teams without reinitiating the program.

Figure 4.8 demonstrates the response given by PROJMNG when it cannot find files for the team number requested in either options 1 or 2 of the monitor's Main Menu. The read cue '?' indicates to the operator that an

input is expected. In the operations demonstrated throughout this thesis '====>' indicates the user input in response to the PROJMNG queries. Figure 4.8 also presents a case in which the program does find the requested files and proceeds to query the monitor for which DP level he desires to evaluate.

```
          WHAT TEAM NUMBER?
          ?
====>2
          RECORDS DO NOT EXIST FOR TEAM  2
          WHAT TEAM NUMBER?
          ?
====>3
          WHAT DECISION POINT DO YOU WISH TO ANALYZE?
          ?
====>3
```

Figure 4.8 Team Number Responses.

The queries illustrated in Figure 4.8 complete the functions of the program routine TEAMIN. The program returns to the main program text and enters the SELECT routine which displays the Selection Menu as it did for the teams, Table XIII. Table XIII functions the same as before except that the entries are not stored in the team's database file. The changes are made only to the monitor's disk data buffer arrays. In option 2 of the monitor's main menu, the changes are also stored on the monitor's copy of the team files in his file listing. These changes are not applied to the PROJMNG disk files unless the monitor specifically sends the changed files to that disk.

In response to option '0' in Table XIII PROJMNG immediately completes the calculation of fee percentages in the program routine PRCTCK. Successful completion of the percentage check is followed with the calculation of achieved values. This process occurs with the program calls to the routines:

TABLE XIII

Selection Menu

```
-----------------------------------------------------------------------------------
  NO.*ITEM******CURRENT VALUE @ DP-3       NO.*ITEM*************CURRENT VALUE
   1. CONTRACTOR            :  1            16. GUIDANCE QUAL TESTS    :  6.00
   2. GUIDANCE APPROACH     :  1            17. FLIGHT TESTS           : 17.00
   3. GUIDANCE CONFIGURATION:  2            18. MAXIMUM COST         M$: 53.00
   4. MAINTAINABILITY ENG M$:  1.75         19. MINIMUM COST         M$: 47.00
   5. VALUE ENG           M$:  0.75         20. MAX COST INCENTIVE    %:  4.00
   6. PARALLEL DEVELOP    M$:  0.0          21. LATEST WEEK            :238.00
   7. MOTOR RELIABILITY    %: 92.00         22. EARLIEST WEEK          :202.00
   8. AIRFRAME RELIABILITY %: 97.00         23. MAX DELIVER INCENTIVE%:  3.50
   9. LAUNCHER RELIABILITY %: 98.50         24. MAXIMUM RELIABILITY   %: 81.00
  10. FIRE CONTROL ERROR YDS: 80.00         25. MINIMUM RELIABILITY   %: 75.00
  11. GUIDANCE ERROR     YDS: 70.00         26. MAX RELIABIL INCENTIV%:  3.50
  12. MOTOR QUAL TESTS      : 25.00         27. MAXIMUM ERROR       YDS:160.00
  13. AIRFRAME QUAL TESTS   :  6.00         28. MINIMUM ERROR       YDS:140.00
  14. LAUNCHER QUAL TESTS   :  3.00         29. MAX ERROR INCENTIVE   %:  4.00
  15. FIRE CONTROL QUAL TSTS:  3.00         30. WEEK FOR LOT 10        :337.00
                  *****0. NONE***********31. CHANGE BY PAGE*****
                SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3.
```

1. ZRW - sets all non-database variables used in the computations to zero.
2. INPUT3 - reads the data tables in files FT17F001, FT18F001, FT19F001 and FT20F001
3. PRESET - uses the database values to make initial adjustments to the achieved value indices
4. TRADE - evaluates the team's input tradeoffs and uses them to generate affects which to apply in calculating the achieved value indices
5. DESRES - degrades the performance factors based on requests for system test.
6. DESRET - degrades the performance for the effects of selected engineering values and performs the calculations for production costs. (This feature, production cost computation, was not used in CNP and has not been activated in PROJMNG.)
7. SE - applies resource parameters to the achieved value calculations.
8. MCD9 - combines the calculation results from the previous three routines to derive a total cost factor for develcpment.
9. RERUN - selectively sets recursive variables to zero to enable the program to finish the calculations of one DP and to reloop to compute the values for successive DP's
10. REPORT - completes the calculations after all DP level computations have been completed.
11. PRCSUM - performs conversion of the achieved values to the appropriate power of ten for display to the user and generates the display, Table XIV.

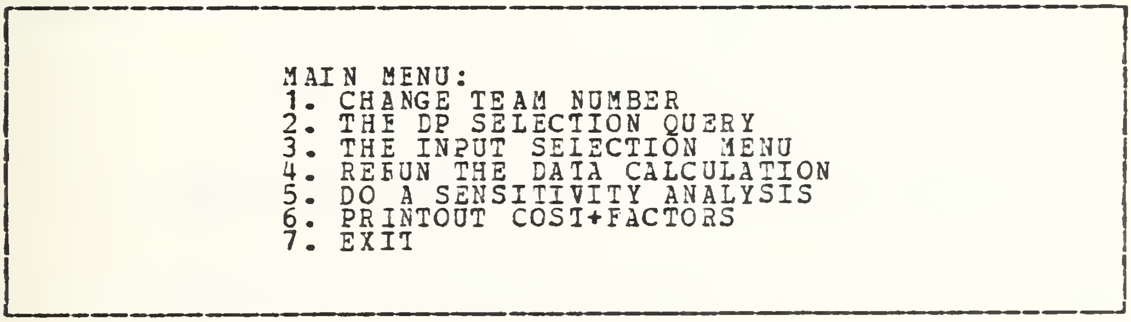
2. Summary of Basic Capabilities

To this point, the basic capabilities of PROJMNG which it has in common with CNP have been discussed. The improved displays, access and queries discussed parallel those features given to the students, with exception of student access to COST + FACTORS. In the next section, the 'flexible menu' feature of the monitor's main menu, and the associated capabilities for sensitivity analysis, for data displays, and for analysis graphing are described.

I. SENSITIVITY ANALYSIS ROUTINES

1. Sensitivity Analysis Basic Menu Options

Figure 4.9 shows again the monitor's menu in mode 1 operation for comparison to the expanded menu which is now described. In order to accomplish the calculation of a sensitivity analysis, option 5 from the menu is selected. In performing this selection, a flag is set 'on' to indicate the availability of values for use in the other displays of the expanded menu options.



```
MAIN MENU:
1. CHANGE TEAM NUMBER
2. THE DP SELECTION QUERY
3. THE INPUT SELECTION MENU
4. RERUN THE DATA CALCULATION
5. DO A SENSITIVITY ANALYSIS
6. PRINTOUT COST+FACTORS
7. EXIT
```

Figure 4.9 Monitor's Main Menu.

Selection of number 5 from the menu immediately accesses the programs routine SNSITIV. This subroutine provides a selection menu (Table XV), queries parameters and parameter data for variation in the analysis, edits the parameter data to remain within DMS scenario limitations, divides the variable parameter into 10 steps for 11 levels of evaluation, runs eleven cycles through the achieved values calculations, displays the table of achieved values at each level, and stores these values in an array for later use in redisplaying the table and for plotting the computations.

Following the display of Table XV, the monitor makes a selection of which DMS data item to analyze. His selection determines a series of questions which PROJMNG displays in order to resolve the analysis parameter details. Figure 4.10 shows one of these queries. In this case the Flight Tests will be varied. The maximum and minimum values permitted are displayed with the queries for range values. These are not only cues to the user but also are the limitations which PROJMNG imposes.

If the selection exceeds the scenario design limits, PROJMNG will take one of two actions. In the first case, PROJMNG will reset the value to be within the range

```
====>14
     INPUT THE VALUES TO VARY FLIGHT TESTS BETWEEN.
     WHAT IS THE LOWER VALUE: >10. ?
     ?
====>10
     WHAT IS THE UPPER VALUE: <25. ?
     ?
====>25
```

Figure 4.10 Flight Tests Demonstration.

TABLE XIV

Development Contract Summary

```
--------------------------------------------------------------------------
DP-3 *** DEVELOPMENT CONTRACT SUMMARY *** TEAM  3

                 INCENTIVE PROVISIONS             INCENTIVE ACHIEVEMENTS

 INCENTIVE      WORST      BEST      MAX FEE     ACHIEVED      FEE       FEE %
   AREA         VALUE      VALUE     ALLOWED      VALUE       EARNED     EARNED

DEV. COST      $ 53.0M   $ 47.0M      4.0%      $ 48.54M    $ 1.486M    2.97%
FLT TST COMPL   238 WK    202 WK      3.5%        199 WK    $ 1.750M    3.50%
RELIABILITY      75.0%     81.0%      3.5%       80.08%     $ 1.480M    2.96%
ACCURACY        160 YDS   140YDS      4.0%       143YDS     $ 1.681M    3.36%

TOTALS                               15.0%                  $ 6.397M   12.79%

TOTAL CONTRACT PRICE               = $ 54.9M
```

TABLE XV

Sensitivity Analysis Menu

```
-----------------------------------------------------------------------
    *****SELECT THE ITEM WHICH IS TO BE SENSITIVITY ANALYZED.*****

     1. MAINTAINABILITY ENG COST  13. GUIDANCE QUAL TESTS
     2. VALUE ENGINEERING COST    14. FLIGHT TESTS
     3. PARALLEL GUIDANCE COST    15. DEVELOPMENT COST RANGE
     4. MOTOR RELIABILITY RANGE   16. DEV COST INCENTIVE %
     5. AIRFRAME RELIABLE RANGE   17. END WK OF FLIGHT TEST RANGE
     6. LAUNCHER/GSE RELIABILITY  18. FLIGHT TEST COMP INCENTIVE
     7. FIRE CONTROL IMPACT ERROR 19. SYSTEM RELIABILITY RANGE
     8. GUIDANCE IMPACT ERROR     20. SYS RELIABILITY INCENTIVE %
     9. MOTOR QUAL TESTS          21. IMPACT ERROR RANGE
    10. AIRFRAME QUAL TESTS       22. IMPACT ACCURACY INCENTIVE
    11. LAUNCHER/GSE QUAL TESTS   23. DEPLOYMENT DATE RANGE
    12. FIRE CONTROL QUAL TESTS
                   *******0. NONE*******
```

permitted. If the second case occurs, the user will be requeried for input. This occurs when PROJMNG doesn't understand the limits requested.

2. Sensitivity Analysis Table

The Sensitivity Table, Table XVI provides a simple column display of the achieved values computed at each step of the analysis as a function of the related parameters. The steps are labeled in the lefthand column. The second column lists the value of the selected parameter for the step. Some of the parameters will result in duplicate steps. PROJMNG divides the range of the parameter values by 10 which can result in a fraction. In creating a step for the table, the parameter value can be successively rounded to the same value. PROJMNG is telling the monitor, it has recognized the parameter as requiring an integer value. It has provided the closest integer to the value. The duplication is the occurrence of the closest integer as the same number on successive occasions. The first occasion PROJMNG rounded up to the value and on the second occasion it rounds down.

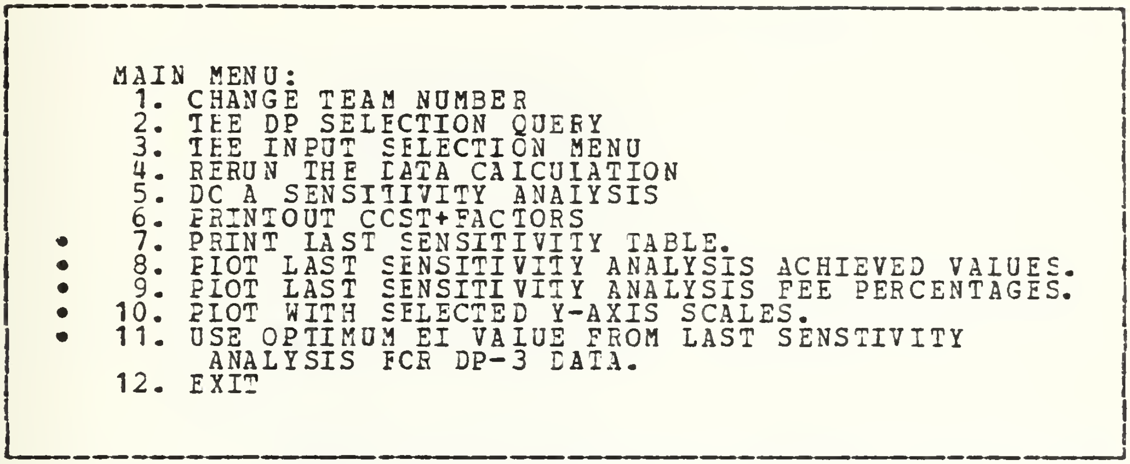
The remaining columns, except EI, are self explanatory. They correlate with the achieved values data given in the table of Development Contract Summary, Table XIV. EI is explained later in this section.

Upon completion of the table, PROJMNG displays the cue of "IHC001A PAUSE ;PRESS <<ENTER>> TO CONTINUE". This holds the display for review by the monitor and reminds the monitor of the action to perform after reviewing the table.

3. Menu Expansion

Pressing <<ENTER>> causes the program to continue. The sequential routing of PROJMNG returns the monitor's MAIN

MENU display in the main program. Figure 4.11 demonstrates the flexible menu with the new options. These options are discussed as we proceed further into PROJMNG.



```
     MAIN MENU:
       1. CHANGE TEAM NUMBER
       2. THE DP SELECTION QUERY
       3. THE INPUT SELECTION MENU
       4. RERUN THE DATA CALCULATION
       5. DO A SENSITIVITY ANALYSIS
       6. PRINTOUT COST+FACTORS
 •     7. PRINT LAST SENSITIVITY TABLE.
 •     8. PLOT LAST SENSITIVITY ANALYSIS ACHIEVED VALUES.
 •     9. PLOT LAST SENSITIVITY ANALYSIS FEE PERCENTAGES.
 •    10. PLOT WITH SELECTED Y-AXIS SCALES.
 •    11. USE OPTIMUM EI VALUE FROM LAST SENSTIVITY
           ANALYSIS FOR DP-3 DATA.
      12. EXIT
```

Figure 4.11 Monitor's Main Menu in Mode 1.

Number 7 on the menu returns to the last display of the Sensitivity Table viewed. PROJMNG's subroutine SNSITIV is reentered through a call (entry) at the end of the subroutine, ENTRY SNSPRT (enter SNSITIV and print the table). SNSITIV's array of stored values enables PROJMNG to reprint the table without any calculations.

4. Sensitivity Analysis Plotting

The second new feature available through the main menu is the capability to graph the table results. The stored values in the array are passed to the PLTSCH subroutine in a data buffer, COMMON PLOTVA. The graph is 20 lines high and fills the entire screen. Consequently a pause statement at the bottom of the graph could not appear on the same screen with the display.

TABLE XVI

Sensitivity Table

```
-----------------------------------------------------------------------------------------
#|SENSITIV|DEVELOPMNT|FLIGHTTEST|RELIABILTY|ACCURACY|FEE|CONTRACT|EI|
 | FLIGHT |FEE| COST |FEE| DATE |FEE|  %   |FEE|YDS | % |  COST  |  |
 | TESTS  | % |      | % |COMPLT| % |      | % |    |TCT|   $M   |  |
_|________|  ACHIEVED | ACHIEVED_|_ ACHIEVED_|ACHIEVED|ACH|ACHIEVED|__|
 0  10.00  4.00 41.22 3.50  197   0.0  73.12 1.36 153   8.86 45.65 80.91
 1  11.00  4.00 42.66 3.50  197   0.0  74.25 1.62 151   9.12 47.22 82.62
 2  13.00  4.00 45.12 3.50  197   0.83 76.42 2.33 148  10.66 50.45 86.03
 3  14.00  4.00 45.95 3.50  197   1.29 77.21 2.65 146  11.43 51.66 87.28
 4  16.00  3.58 47.62 3.50  199   2.37 79.06 3.15 144  12.61 53.93 89.57
 5  17.00  2.97 48.54 3.50  199   2.96 80.08 3.36 143  12.79 54.94 90.57
 6  19.00  1.75 50.38 3.50  200   3.50 81.66 3.85 140  12.59 56.68 92.27
 7  20.00  1.15 51.28 3.40  203   3.50 82.61 4.00 139  12.05 57.30 92.88
 8  22.00  0.17 52.74 2.63  211   3.50 84.08 4.00 137  10.30 57.89 93.44
 9  23.00  0.0  53.96 2.04  217   3.50 84.90 4.00 137   9.54 58.73 94.24
10  25.00  0.0  55.47 0.88  229   3.50 85.31 4.00 136   8.38 59.66 95.12
IHO001A PAUSE ;PRESS <<ENTER>> TO CONTINUE.

 IHC001A PAUSE ;PRESS <<ENTER>> TO CONTINUE.
```

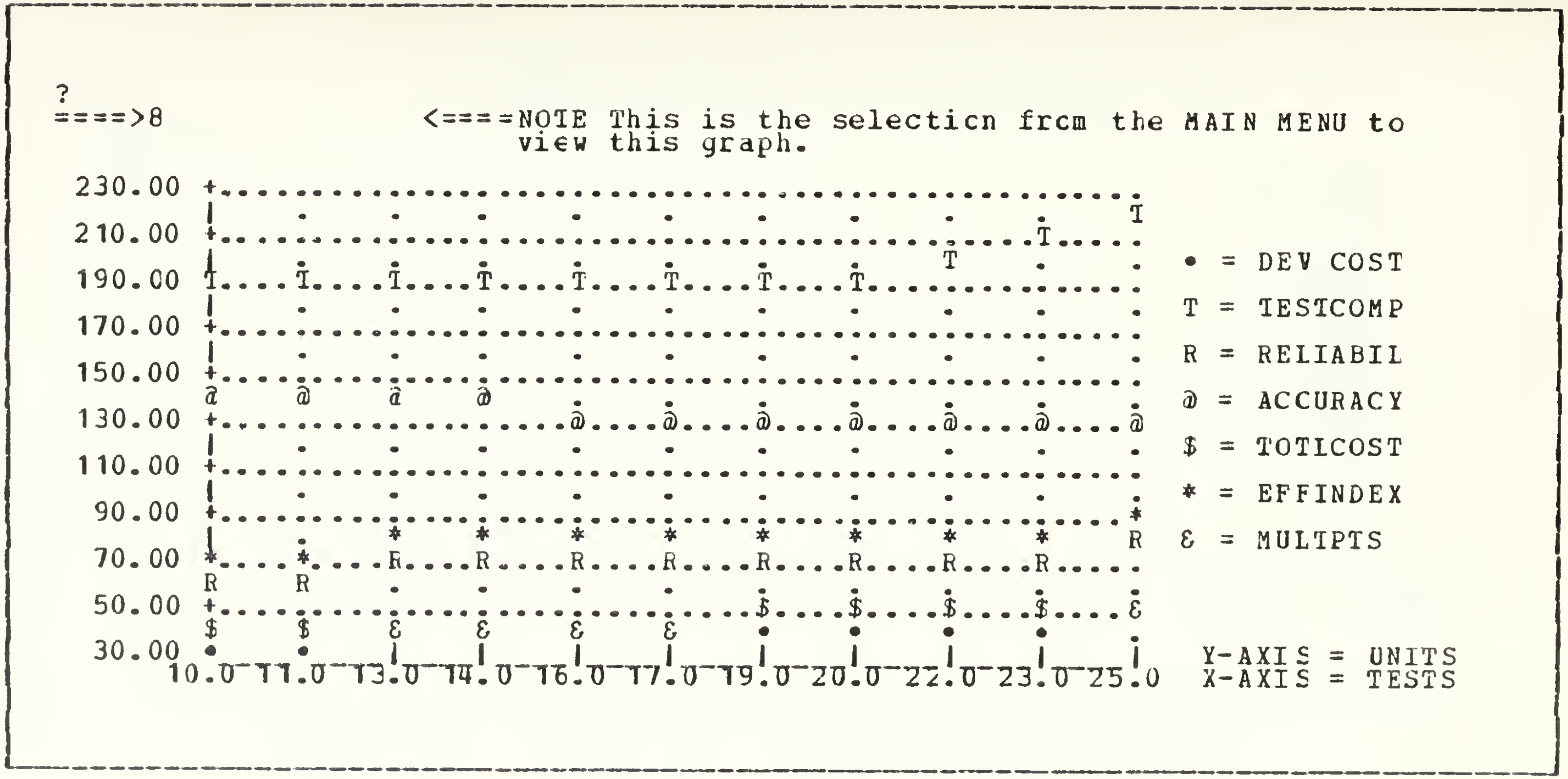
In order to alleviate this difficulty the pause has been eliminated. The display appears on one screen with the addition of a blank line which forces the IBM to hold the display under the condition of a screen overflow, as indicated by the "MORE" in the screen lower right corner. The screen can be held only temporarily in this IBM display mode. If the operator desires to extend the hold for closer analysis of the graph, pressing <<ENTER>> will change the IBM display mode to "HOLDING" which can remain indefinitely. Once the operator has finished with the "MORE" display mode, or the "HOLDING" mode he can continue by clearing the screen. On the IBM 3278 this is accomplished by pressing <<ALT>> and <<CLEAR>> simultaneously.

The definition key for characters is displayed on the graph, as given in Figure 4.12. A series of print statements has been place in PLTSCH to list the key in the righthand graph margin.

```
• = DEVELOPMENT COST
T = FLIGHT TEST COMPLETION DATE
R = RELIABILITY
@ = ACCURACY
% = TOTAL FEE PERCENTAGE
$ = TOTAL COST
* = EFFECTIVENESS INDEX
& = Several curves intersect
```

Figure 4.12 Plot Character Key Definitions.

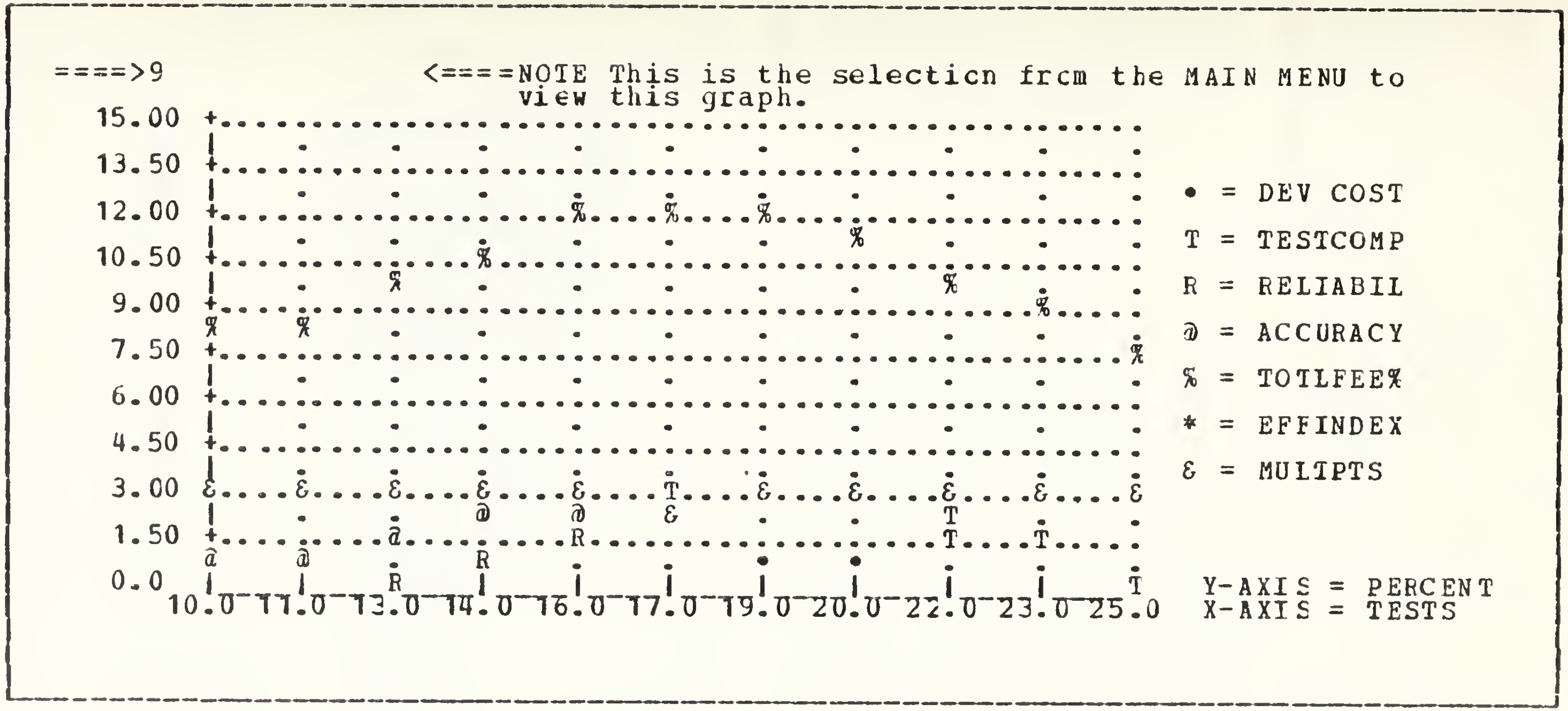
The table displayed by "8. PLOT LAST SENSITIVITY ANALYSIS ACHIEVED VALUES." divides the screen height of twenty lines into the largest value which is to be displayed. This functions as the ordinate scale for the graph, Figure 4.13. PLTSCH displays the entire set of data. In the next selection, "9. PLOT LAST SENSITIVITY ANALYSIS



```
?
====>8                  <====NOTE This is the selection from the MAIN MENU to
                             view this graph.
```

```
• = DEV COST
T = TESTCOMP
R = RELIABIL
@ = ACCURACY
$ = TOTLCOST
* = EFFINDEX
& = MULTPTS

Y-AXIS = UNITS
X-AXIS = TESTS
```

Figure 4.13 Sensitivity Table.

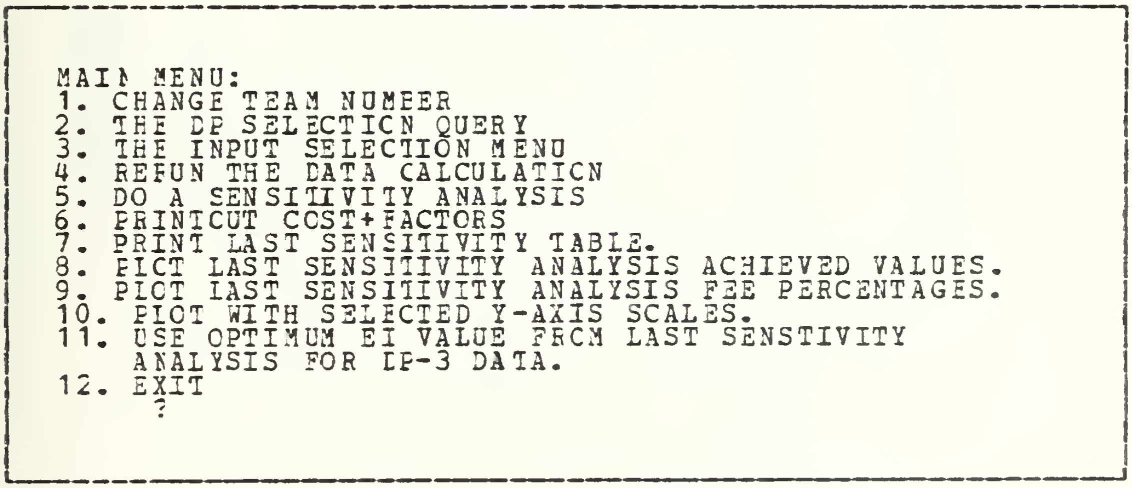


```
====>9              <====NOTE This is the selection from the MAIN MENU to
                         view this graph.
15.00 +..................................................
      |    .    .    .    .    .    .    .    .    .    .
13.50 +..................................................
      |    .    .    .    .    .    .    .    .    .    .   ● = DEV COST
12.00 +....................%....%....%....................
      |    .    .    .    .    .    .    %    .    .    .   T = TESTCOMP
10.50 +..............%...................................
      |    .    %    .    .    .    .    .    %    .    .   R = RELIABIL
 9.00 +.............................................%.....
      %    %    .    .    .    .    .    .    .    .    .   @ = ACCURACY
 7.50 +.................................................%
      |    .    .    .    .    .    .    .    .    .    .   % = TOTLFEE%
 6.00 +..................................................
      |    .    .    .    .    .    .    .    .    .    .   * = EFFINDEX
 4.50 +..................................................
      |    .    .    .    .    .    .    .    .    .    .   & = MULTPTS
 3.00 &....&....&....&....&....T....&....&....&....&....&
      |    .    .    @    @    &    .    .    T    .    .
 1.50 +.........@.........R...................T....T.....
      @    @    .    R    .    .    ●    ●    .    .    .
  0.0 |    |    R    |    |    |    |    |    |    |    T    Y-AXIS = PERCENT
    10.0-11.0-13.0-14.0-16.0-17.0-19.0-20.0-22.0-23.0-25.0   X-AXIS = TESTS
```

Figure 4.14 Sensitivity Analysis Percentages Graph.

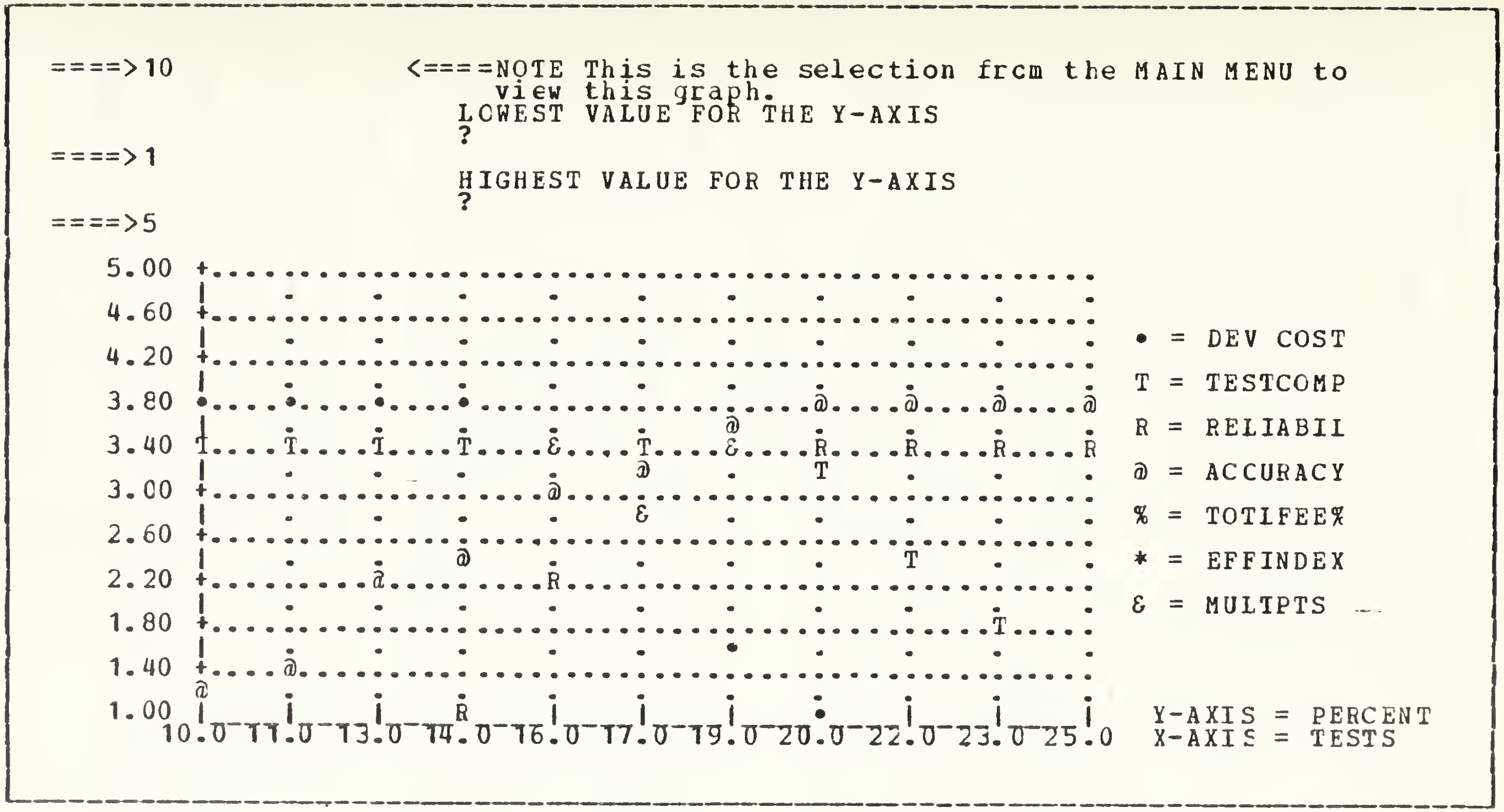
FEE PERCENTAGES." the vertical axis is provided with a scale of 0.5 per line. The height of the ordinate becomes 10, which will display the fees up to 10%. The benefit of this change in scales is in the presentation of a magnified view in the y-axis between the range of 0 and 10 where the fee percentages are concentrated. Figure 4.14 demonstrates the percentage graph.

Graphing can also be scaled by the monitor. Choosing Sensitivity Menu option number 10 directs PROJMNG to ask for the ranges which are to be used for the y-axis. Figure 4.15 demonstrates the interactive exchange between the program and user to create the desired y-scale. The program accepts any values for the minimum and maximum values for the axis. The program again divides this range into 20 steps. The step size becomes the y-axis scale. If the scale is less than 0.05 per line, PROJMNG resets the scale to 0.05. Figure 4.16 demonstrates the graph resulting from the steps in figure 4.15.



```
MAIN MENU:
1. CHANGE TEAM NUMBER
2. THE DP SELECTION QUERY
3. THE INPUT SELECTION MENU
4. RERUN THE DATA CALCULATION
5. DO A SENSITIVITY ANALYSIS
6. PRINTOUT COST+FACTORS
7. PRINT LAST SENSITIVITY TABLE.
8. PLOT LAST SENSITIVITY ANALYSIS ACHIEVED VALUES.
9. PLOT LAST SENSITIVITY ANALYSIS FEE PERCENTAGES.
10. PLOT WITH SELECTED Y-AXIS SCALES.
11. USE OPTIMUM EI VALUE FROM LAST SENSTIVITY
    ANALYSIS FOR DP-3 DATA.
12. EXIT
     ?
```

Figure 4.15 Monitor's Main Menu in Mode 1.



```
====>10            <====NOTE This is the selection frcm the MAIN MENU to
                       view this graph.
                     LOWEST VALUE FOR THE Y-AXIS
                     ?
====>1
                     HIGHEST VALUE FOR THE Y-AXIS
                     ?
====>5
     5.00 +..................................................
          |    .    .    .    .    .    .    .    .    .    .
     4.60 +..................................................
          |    .    .    .    .    .    .    .    .    .    .    • = DEV COST
     4.20 +..................................................
          |    .    .    .    .    .    .    .    .    .    .    T = TESTCOMP
     3.80 •....•....•....•...................@....@....@....@
          |    .    .    .    .    .    @    .    .    .    .    R = RELIABIL
     3.40 T....T....T....T....ε....T....ε....R....R....R....R
          |    .    .    .    .    @    .    T    .    .    .    @ = ACCURACY
     3.00 +...................@..............................
          |    .    .    .    .    ε    .    .    .    .    .    % = TOTLFEE%
     2.60 +..................................................
          |    .    .    @    .    .    .    .    T    .    .    * = EFFINDEX
     2.20 +.........@.........R..............................
          |    .    .    .    .    .    .    .    .    .    .    ε = MULTPTS
     1.80 +............................................T.....
          |    .    .    .    .    .    •    .    .    .    .
     1.40 +....@.............................................
          @    .    .    .    .    .    .    .    .    .    .
     1.00 |    |    |    R    |    |    |    •    |    |    |     Y-AXIS = PERCENT
        10.0‾11.0‾13.0‾14.0‾16.0‾17.0‾19.0‾20.0‾22.0‾23.0‾25.0    X-AXIS = TESTS
```

Figure 4.16 Selected Vertical Axis.

5. Effectiveness Index

Selection 11 on the Sensitivity Menu provides the monitor with a means of directly copying into the data array (Selection Menu) the best sensitivity parameter value for the highest EI value. SNSITIV determines the best occurrence of the highest value of Effectiveness Index. The EI with the lowest parameter value is used for cost analysis, accuracy analysis, and completion date analysis. For parameters involving 'reliability', the EI of the highest parameter value is stored in EIOPT. It will not make a change of the optimum selection stored in those cases where the minimum value is zero. It stores the sensitivity parameter's step number in variable EIOPT. When option 11 is selected, the sensitivity parameter being evaluated has its value changed to that of the evaluation step which had been saved in EIOPT. Figure 4.17 demonstrates the revised Selection Menu.

6. Flexible Menu Reset

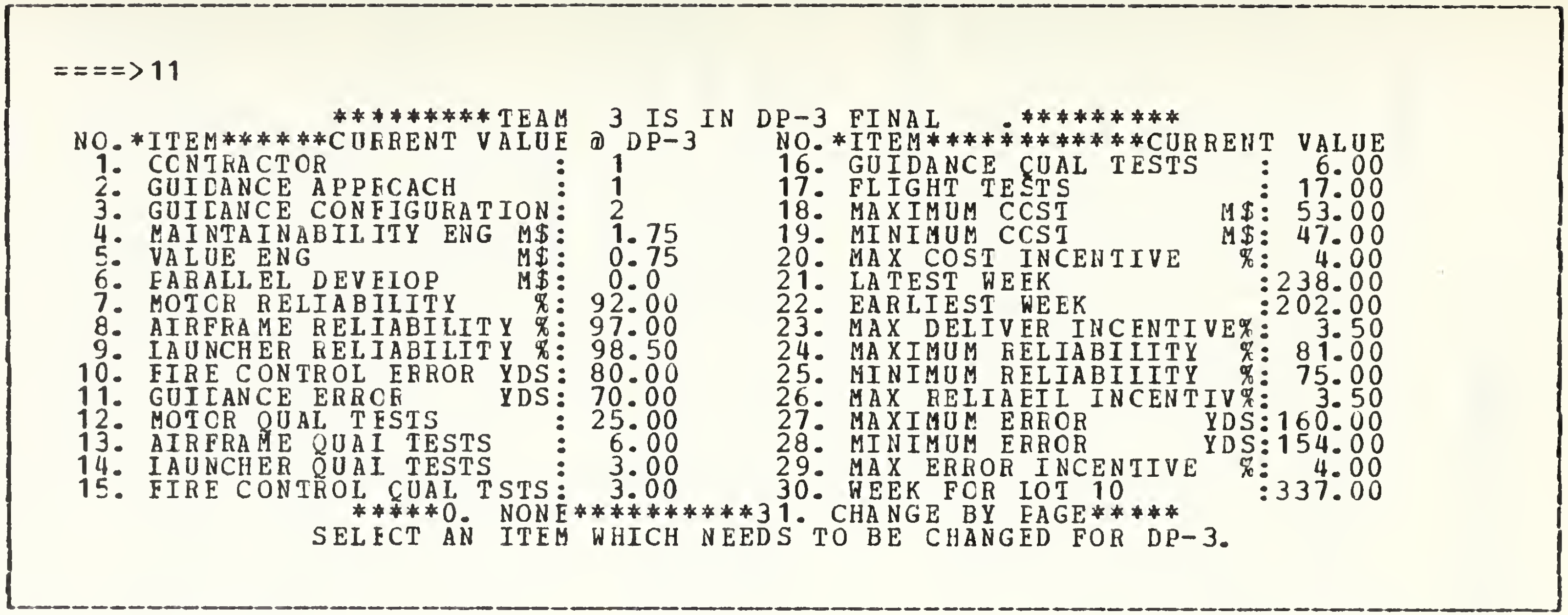
Should the monitor change the team, the DP level or the data in the database the calculations made by SNSITIV are no longer applicable. PROJMNG flags the use of the subroutines which generate these changes. Through the running of these subroutines, the flag which expands the Sensitivity Menu is reset to 'off'. The option to replot graphs or tables based on the previous set of data is invalidated.

Selection of the mode 2 option to "FILE TEAM PROPOSAL" generates the monitor mode flag MR. For mode 2, MR=2. This flag causes the menu statement to file the proposal, and flags the program when leaving mode 2 either at exit or on return to the Monitor's Mode Menu to ask if the rewritten file should be sent to the PROJMNG disk.

J. SUMMARY

The PROJMNG FORTRAN program retains its basic computations and organization in common with the original CNP program. These basic features have been enhanced by improved user/program interaction, and by the reformulation of the achieved value fee percentage computation algorithm.

PROJMNG FORTRAN has an additional benefit over CNP in the extended sensitivity analysis capability it provides to the monitor. Both the sensitivity tables and graphs enhance the monitor's ability to evaluate the team's progress when using PROJMNG.



```
====>11

            **********TEAM  3 IS IN DP-3 FINAL   .**********
 NO.*ITEM******CURRENT VALUE @ DP-3    NO.*ITEM**************CURRENT VALUE
  1. CONTRACTOR              :  1       16. GUIDANCE QUAL TESTS    :  6.00
  2. GUIDANCE APPROACH       :  1       17. FLIGHT TESTS           : 17.00
  3. GUIDANCE CONFIGURATION:   2       18. MAXIMUM COST         M$: 53.00
  4. MAINTAINABILITY ENG M$:  1.75     19. MINIMUM COST         M$: 47.00
  5. VALUE ENG           M$:  0.75     20. MAX COST INCENTIVE    %:  4.00
  6. PARALLEL DEVELOP    M$:  0.0      21. LATEST WEEK             :238.00
  7. MOTOR RELIABILITY    %: 92.00     22. EARLIEST WEEK           :202.00
  8. AIRFRAME RELIABILITY %: 97.00     23. MAX DELIVER INCENTIVE%:  3.50
  9. LAUNCHER RELIABILITY %: 98.50     24. MAXIMUM RELIABILITY   %: 81.00
 10. FIRE CONTROL ERROR YDS: 80.00     25. MINIMUM RELIABILITY   %: 75.00
 11. GUIDANCE ERROR     YDS: 70.00     26. MAX RELIABIL INCENTIV%:  3.50
 12. MOTOR QUAL TESTS      : 25.00     27. MAXIMUM ERROR       YDS:160.00
 13. AIRFRAME QUAL TESTS   :  6.00     28. MINIMUM ERROR       YDS:154.00
 14. LAUNCHER QUAL TESTS   :  3.00     29. MAX ERROR INCENTIVE   %:  4.00
 15. FIRE CONTROL QUAL TSTS:  3.00     30. WEEK FOR LOT 10         :337.00
              *****0. NONE**********31. CHANGE BY PAGE*****
           SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3.
```

Figure 4.17   Selection Menu with Optimized Flight Tests.

V. SUMMARY AND CONCLUSIONS

A. SUMMARY

The DMS exercise provides a time proven means of educating project managers in the potentials, sensitivities and incentives of acquisition management. This same capability is the significance of CNP and of PROJMNG as training aids in the DMS exercise. The support of project management training by computer programs has existed for the past twenty years. This thesis has upgraded the available computer program support for DMS. PROJMNG provides DMS with a user-friendly interactive tool to assist in the aquisition management training process.

B. DMS

DMS has proven to be an effective training exercise. It provides personnel in the Department of Defense, and personnel in other system acquisition environments, with relevant management training.

1. Benefit

The benefit of DMS as a training exercise is in the requirement it places on students to understand acquisition management concepts. It provides a sequenced set of events to develop, produce and deploy a missile The Decision Point's serve as planning milestones that the student teams must prepare for in order to successfully manage their acquisition project. Their understanding of the concepts of acquisition management is challenged by the exercise in a pseudo-real time scenario.

2. Computer Support

During the twenty years in which DMS has been available, there has been a revolution in the tools available to support project management. The ability of computers to rapidly organize, process and correlate data has changed the acquisition management environment.

3. CNP

DMS has adapted somewhat to management evolution by its incorporation and development of CNP.

CNP is an interactive Fortran program which has been used for the past decade. In its current configuration, it is difficult to use. Since it is not user friendly.

C. PROJMNG

In order for DMS to remain a relevant management training exercise, it must be kept current in its state-of-the-art management concepts, training facility, and computer support. The Fortran program PROJMNG has been created to provide DMS a program which is more user-friendly. It is not just an off-shoot of CNP. It incorporates portions of CNP by virtue of utilizing most of the subroutines and algorithms of CNP. PROJMNG improves the Fortran support for DMS in its refinement of CNP and by expanding the programmed support for analysis.

1. Benefits

PROJMNG's benefits include:

- program access
- program exit
- session record
- data file of input parameters
- menus for operation selection

- expanded sensitivity analysis
- tables and graphs to study parameter sensitivity
- user-friendly displays
- monitor-team communication interaction through datafile exchange

2. Shortcomings

PROJMNG has not been completely validated. Its achieved values have not been tested sufficiently to determine its ability to react in a pseudo-realistic manner to parameter variations.

Additionally, the programming facilities utilized in PROJMNG are not as modern as they could be. The PROJMNG capability has been hampered by the limitations of the IBM-3033. Areas in which these limitations have impacted PROJMNG are discussed in Chapter 6, Recommendations.

3. First Usage

PROJMNG has undergone a field test. It was used to support the Winter 1984 course in Project Management, AS-3501, taught at the Naval Postgraduate School by Lcdr. J. Ferris. This field test resulted in the following program enhancements:

- Ability to mass purge data files
- Ability of the monitor to send files to the PROJMNG disk
- The SMSG autolog feature operates only when no users are accessed to the PROJMNG disk
- All displays are formatted for 80 character wide personal computer (PC) display.
- All exec query responses are tested for input errors.
- PROJMNG EXEC tests files for format before replacing the team disk copy. The file used

for the team database is sequentially the first one which exhibits the correct format from either the PROJMNG disk or the team, and defaults to the file "FILE DATAFILE".

During this field test, the major problem in facilitating the program occurred in the autolog procedure. The SMSG automatic log-on of the PROJMNG disk occurred regardless of users being on the disk. This situation resulted in files being read into the disk without proper organization. The files subjected to this garbling were unusable.

SMSG is now programmed to write the team files onto the PROJMNG disk only when the disk is not accessed by any user.

4. Transportability

The wide usage of DMS in the U. S. and other countries provided interest in making PROJMNG capable of use on various computer systems. CMS unique features and IBM extentions to fortran have been discussed in this thesis to enable adaptation. In its present form PROJMNG relies heavily on CMS and its executive language facilities to manipulate the data files outside of the Fortran program environment. These facilities require assessment on a case by case basis in order for PROJMNG to be usable on any system other than IBM's CMS with the FRTCMS extention.

a. FORTRAN

The program has been compiled both on the FORTRAN H and FORTRAN GI compilers of the NPS IBM-3033. PROJMNG is unable to compile in WATFIV (WATERLOO FORTRAN). The complication which precludes the WATFIV compilation is the call to FRTCMS. It is desirable to not remove the FRTCMS calls. They provide a major enhancement to the program by

enabling the program to clear the screen, 'CLRSCRN '. Loss of this feature would disorganize the PROJMNG displays.

D. DOCUMENTATION

1. CNP

A major problem in working with a program is its documentation. CNP has several styles of documentation. There are periodic lines of text to label or explain the programs operation. These comments were beneficial in reconstructing some of the program operation, but inadequate in deciphering the program operation.

The best documentation present in CNP were the names selected for variables. Each of the input variables, a majority of the computational variables and the routine names were carefully selected to provide a relation with their data definition or with the process they represent. The variables in most of CNP are acronyms.

2. PROJMNG

PROJMNG contains additional comments throughout the program code taken from CNP. Frequent comments have been added to explain the processes accomplished by groups of code and routines. All of the terms added to the program by PROJMNG are meaningful acronyms for the routines and variables.

3. Glossary

Appendix H provides a glossary of the terms used in both CNP and PROJMNG. Compilation of the glossary provides a documentation method which supplements the benefit obtained from selecting variables which are acronyms for the content definition. Some definitions for CNP terms in the glossary remain undetermined.

4. Debugging

Terms used in a program without adequate documentation may interact in unexpected areas of the program. With 3063 variables in the program, CNP is a maze of connections between variables. Tracking the details of terms used in CNP was difficult. The two worst aspects of terms in CNP are multiple definitions for a term and manipulation of the variables to change the value's power-of-ten multiple. For example, RATIO is used in CNP to determine the percentage of a fee to be awarded, it is also used as a value of millions of dollars paid for fees, and it is used in a resource ratio calculation. An example of the manipulation of variables is the array CTC(8). This array is used in millions of dollars and in dollars.

The disadvantage in using mixed definitions for a variable is the impact which occurs on calculations which depend on the variable. If a calculation is based on a specific multiple of the variable and it has varying values dependent on which subroutine used it last, the programmer cannot utilize the variable. He must consider the inability to predict the occurrence of changes to the variables definition.

VI. RECOMMENDATIONS FOR FUTURE CONSIDERATION

The planning, design and execution of this revision to CNP have generated the following recommendations.

- Increase the program support to the exercise
- Expand the program to support more Decision Points in the DMS exercise
- Develop a DMS computer assisted tutorial
- Validate the achieved values
- Change the data tables to formulas
- Increase the sensitivity analysis capabilities
- Improve the graphics resolution
- Change the menu displays to 'plate' displays
- Develop a program to create project management exercise scenarios for actual system acquisitions
- Use PROJMNG to encompass other CNP program variations

A. INCREASED EXERCISE SUPPORT

1. Discussion

The wide availability of personal computers provides project managers and students with improved data processing capability. They now have computation capability available to them instantaneously throughout the acquisition process for storing and analyzing their own specific acquisition data. If DMS is to continue to provide project managers with training at the level now available in the management environment, it must have software facility equivalent to at least that of their personal systems. It is feasible to make the DMS program compatible with PCs. DMS must be supported

at the state-of-the-art. It must be contemporary in providing a program which has the degree of computer facility which the managers experience in day-to-day use and in actual acquisition planning. As this capability develops, the program supporting DMS should expand to:

- Support all DMS Decision Point levels
- Interface with student and manager PCs
- Support actual acquisition scenarios

2. Recommendation

Design new program capabilities to permit DMS adaptation for broader application to PC's and to change the acquisition management process.

B. INCLUDE ALL DECISION POINTS

1. Discussion

Both CNP and PROJMNG support the DMS exercise only for DP-3 and DP-4. The present DMS scenario includes six Decision Points. All of the DP's require team analysis of the exercise parameters and submission of Decision Sheets (see Appendix B). These characteristics are conducive to incorporation of these DP's into other PROJMNG like analyses.

Use of an interactive computer program to support DP's 1 and 2 would provide an opportunity for each team to do a better analysis in these early Decision Points, and to initiate the team database for later use. Inclusion of all DP's in the program would permit documentation of the entire team exercise performance in one computer record.

2. Recommendation

A future computer support program for DMS should be designed to expand exercise support from the PROJMNG and CNP capability of DP 3 and 4 evaluation to all exercise DP's.

C. DEVELOP A DMS TUTORIAL

1. Discussion

Computer tutorials have become a popular method of training. The availability of computer assets for CNP or PROJMNG indicates the presence of computer hardware to support a tutorial program. Student interaction with a computer supported DMS tutorial could be used to introduce the student to the DMS exercise, to perform sample problems, and to give examples.

The one-on-one capability for student reponse to a DMS tutorial should be limited to exercise orientation and familiarization. DP's 2 through 6 provide an opportunity for a team-work experience. This team interaction should remain an integral part of the DMS exercise.

2. Recommendation

Incorporate the DMS training scenario into an acquisition management tutorial. The tutorial is seen as providing:

- scenario presentation to the individual
- review of initial positions
- aid to students to remedy difficulty with concepts

D. VALIDATE PROJMNG

1. Discussion

Correlation of achieved values between the CNP/PROJMNG programs and the DMS data values has not been demonstrated. An analysis should be conducted in order to determine if the algorithms in PROJMNG provide realistic approximations to the full DMS batch program.

Evidence exists of differences between some CNP and PROJMNG results. Either program may be at fault. PROJMNG's errors result from its acceptance of the algorithms and calculations contained in the original CNP program. The values computed by CNP can not be assumed to be accurate.

Current classroom usage of PROJMNG has been predicated on the assumption that these values indeed perform as should be expected. This assumption was based on the incorporation of the Fortran code for the achieved values calculation from CNP into PROJMNG.

2. Recommendation

A series of evaluations should be conducted to compare the actual DMS batch outputs and with the performance of PROJMNG.

E. REPLACE THE DATA TABLES WITH EQUATIONS

1. Discussion

The tables of values included in CNP/PROJMNG have limited range in which performance data can be obtained. Beyond these limits data values are not available for calculations. In most cases, CNP does not warn the user that he has performed a calculation which is not a feasible solution. In most cases, PROJMNG forces the team to resubmit data items which are outside the range capabilities

discussed in the DMS instructions. There are parameters for which limits are not specified in the DMS handouts, and which can result in unusual calculations such as achieving a flight test completion date of 0.

There are two reasons for redesigning the program resource value algorithms which are used to calculate achieved values. First is to provide a continuous function instead of a step function. The table values read for calculations in both CNP and PROJMNG do not provide continuous value functions. Second the values have a limited range. Student selections which are beyond the capability of the table result in erroneous achieved values, such as the flight test example above.

The results provided by CNP and PROJMNG, as discussed in Chapter 1, are not actual results. They represent approximate results to reflect the real world issue of contract risk.

The programming of equations to obtain the resource values will result in the need to validate the achieved values predictions. Validating the values achieved as a result of changing the resource allocation algorithm would be of equal complexity to validating PROJMNG's current results. Both validation projects would verify the program's capability to produce realistic predicted contract performance values. In the interest of reducing the expenditure of effort between these two validations, the effort would be best applied in one project. To validate PROJMNG and later require revalidation for a new algorithm would obviate the former.

2. Recommendation

First, redesign and implement the equations for determining resource allocations. Second, validate the entire new program package of PROJMNG for DMS support.

F. INCREASE SENSITIVITY ANALYSIS OPTIONS

1. Discussion

The options available in the sensitivity analysis menu do not have as dramatic an effect in demonstrating the impact of incentives and other parameters as had been originally expected. Sensitivity curves showing definite peaks or valleys in the achieved effectiveness index are not demonstrated as readily as had been anticipated. The graphic presentation and sensitivity selection options should be designed to provide the user with a clear depiction of the effect of incentive variation. The user should be able to readily determine from the sensitivity analysis products the benefits or disadvantages of changing his decision above or below his current position.

2. Recommendation

Determine other analysis options which may have greater value to the monitors. Conduct a series of analyses to determine if they affect the sensitivity, and to eliminate those options which do not cause an achieved value fluctuation.

G. IMPROVED GRAPHICS

1. Discussion

One area of computer hardware and software development which is rapidly changing is graphics. The graph subroutine in PROJMNG was specifically designed for the program and has a number of limitations. No other IBM-3033 graphing program was found which provides multiple curve graphing, selectable scales and the number of variations which PROJMNG provides. A (continuous) curve plot is needed to enhance demonstration of parameter variations. The

graphic detail which is needed to improve the display does not appear to be within the capability of the IBM-3278 terminals which are currently a widely available computer peripheral for student use, and which are particularly available in large quantities at NPS. What is needed is the ability to control screen display for continuous line plots. This type of resolution for the graphics on the IBM-3033 is not within the IBM-3278 terminals capability. Terminals do exist which provide greater screen control and which can be programmed to plot continuous lines for the IBM.

Display detail is undergoing evolution. The advantages of color for highlighting presentations adds a perceptible benefit in human ability to comprehend. Color can be used to add a perception of relative weight of significance between curves and between points on a curve.The use of characters and figure presentations which exceed the standard typewriter characters in ASCII presentations increase the pictorial communication of data. Both of these factors -color graphics and detailed display pixle control- are receiving widespread attention in interactive computer systems and should be considered as enhancements for computer support programs which will be successors to CNP and PROJMNG.

A benefit to plotting continuous curves is the ability to separate points of curve intersection. Another benefit is the ability of continuous curves to show the connection between points in the same curve. The PROJMNG plots readily demonstrate the difficulty in presenting the different curves whose plots have a coinciding point.

In the short term, use of Fortran capable displays on the Tektronix 618 or the Ramtek does not appear reasonable. Several factors degrade the significance of implementing operation on a more complex graphic system. One is the added processing time and cost for a display. The

second reason to not consider graphing the program on a graphics display system is the rapid rate of technical development currently occurring. Based on the presence of new IBM color graphics terminals at NPS, the facility at the W. R. Church Computer Center may be expected to provide high resolution color graphics for users in the not to distant future. In this case the effort to recode the PROJMNG program for high resolution graphics on the Tektronix or the Ramtek graphics terminals would be obviated both in comparative cost and in utility.

Another factor which should be considered before programming DMS support on a graphics unique system, such as DISSPLA, is whether the system will allow DMS to remain transportable. CNP and PROJMNG were designed to be cost-effective by their potentially wide distribution and usage. The common availability of the systems on which these two programs are based permits them to be widely copied. Use of unique display systems would reduce the tranportability and preclude spreading the cost over a wider user population.

2. Recommendation

Future computer programs to support DMS should demonstrate multi-color plotting capability and continuous line graphing.

H. PLATE DISPLAYS

1. Discussion

The Selection Menu display sets a pattern of presentation of data throughout the PROJMNG program. Menu heading and columns of data labels remain consistent in the monitor modes of operation, in the student mode of operation and in the display of data for the report submission tables. With the capability to produce plate displays, these factors

could be set into a file to be overlaid onto the screen. Items in the table are excellent candidates for a plate information display. The plate display capability would enable data item value change on the screen without reprinting the display and without the need to indicate which item is to be changed. The user will be able to place the terminal curser on the data item value, change the values as desired, and submit the entire change as one input.

Not only would the plate display simplify data change and operation, it would also eliminate the need for four of the subroutines used in PROJMNG. The two change routines, the PAGEIN subroutine and the FINISH subroutines could be combined into one menu with the Selection Menu format.

Plate display capability would also be useful for the Sensitivity Analysis Menu. The format of this display could be changed into the same format as the Selection Menu. Much of the sensitivity analysis routine could be eliminated with the saving in lines-of-code which the plate display represents.

The IBM CMS II version currently being implemented at the Naval Postgraduate School (NPS) contains this capability. Plate display capability for this program is expected to be available within the coming academic quarter.

2. Recommendation

Adapt PROJMNG to display plates as soon as CMS II is implemented.

I. A SCENARIO GENERATING PROGRAM

1. Discussion

Within the computer simulation environment, conceptual level programs have been demonstrated which permit the user to create his own decision support and analysis scenario. Application of this technology to DMS is envisioned as providing a program which can simulate any actual acquisition scenario. These concept level programs utilize a hierarchy of menus to query the user for arrays to establish the program dimensions. The program then queries the user for array size, array name, variables names, parameter ranges for the variables and relative significance (weight) of the variables within the parameter. A hierarchy of menu-driven routines guides the user to create a program which supports the scenario to be studied. These programs enable the user to vary input variables in a matrix of parameters. The user constructs a scenario based on his concept of the parameters, on his feeling of their interaction and their relative significance. Based on these user perceptions such a program creates predictions for the effects of the parameter interaction. These predictions are heavily weighted by user bias to parameter interaction. Usually the program results predict what the user envisioned because of the impact of his biases. The main benefit of scenario creation has proven to be the opportunity for the user to conduct a detailed management study of the situation being modelled.

2. Recommendation

Develop a program whose design would permit project managers to create program simulations tailored to assess their specific real-world scenarios.

J. USE PROJMNG TO ENCOMPASS OTHER CNP VARIATIONS

1. Discussion

The design of PROJMNG around CNP routines with its own routines performing the features which make it different from CNP makes it possible for PROJMNG to encompass other CNP variations. Any CNP style program which has the same common buffer of data could utilize PROJMNG's enhancements. The buffer of common variables in PROJMNG has been editted from that contained in CNP. The CNP buffer can be restored into PROJMNG, and does permit PROJMNG operation.

The design of PROJMNG routines to support the calculations obtained in CNP suggest the potential of PROJMNG as a set of routines to enhance the user-friendliness of other programs. Programs which could generate the same data items as CNP could be used as a nucleus program whose data would be used by PROJMNG routines to conduct sensitivity analysis.

Programs which are candidates for consideration are:

- the main DMS batch game
- the latest ICAF DMS version (SAFE 84)
- the Swedish version called PMG/ZEBRA

2. Recommendation

Investigate the use of PROJMNG as is or modified to be compatible with the latest ICAF DMS version (SAFE 84) and the Swedish version called PMG/ZEBRA. Test the concept of using PROJMNG as an enhancement to the other management acquisition programs.

APPENDIX A

CONTRACT NEGOTIATION PACKAGE (CNP)

Instructions For Using CNP

DP-3 & DP-4

```
         *****************************************************
         *        DO NOT TYPE DATA WHICH ARE IN BOXES        *
         *              ON DECISION SHEETS                   *
         *****************************************************
        ----------------------------------------------------------
        |       Type in the data as shown by arrows(====>)       |
        |--------------------------------------------------------|
        |                DP-3                 |      NOTES       |
        |-------------------------------------|------------------
        |        Log onto the computer        |
        ---------------------------------------

        ---------------------------------------
        | R; T=0.01/0.01 17:58:49             |
====>LINK 0543P 191 192                        LINK TO CNP PROGRAM
        | DASD 192 FORCED R/O; R/W BY 1681P   |
        | R(00102); T=0.01/0.01 17:59:02      |
====>ACC 192 C
        | C (192) R/O                         |
        | R; T=0.01/0.01 17:59:08             |
====>CNP                                       CALLS CNP
        | EXECUTION BEGINS...                 |
        | TYPE DP NUMBER 3 OR 4               |
====>3
        | THIS IS DP NUMBER ***** 3 *****     |
        | Y OR N                              |
====>Y
        | TO CONFIRM, TYPE 1 FOR OK, 0 IF     |
        | CORRECTION MUST BE MADE.            |
        | TEAM NUMBER   XXX                   |
====>003
        | RECEIVED - TEAM NUMBER:    3        |
        | CONFIRM:                            |
====>1
        | NEW OR CHANGES TO INPUT DATA        |  FIRST TIME YOU MUST TYPE ALL 1'S
        |   RETYPE WHOLE PAGE CONTAINING ONE  |
        | 0 (ZERO) FOR NO - 1 (ONE) FOR YES   |
        | TYPE IN COMBINATION, 4 0S AND 1S FOR|
        | PAGES 1 to 4                        |
====>1111
        | TYPE PAGE 1                         |
====>112017500750000
        | RECEIVED FOR PAGE 1:                |
        |     1                               |
        |     1                               |
        |     2                               |
        | 1.75                                |
        | 0.75                                |
        | 0.0                                 |
        | CONFIRM:                            |
====>1
        |                                     |
        |                                     |
        |                                     |
```

```
                | TYPE PAGE 2:
====>92097098500702563617
                | RECEIVED FOR PAGE 2:
                | 92.0
                | 97.0
                | 98.5
                |   80
                |   70
                |
                |   25
                |    6
                |    3
                |    6                                    <====ERROR MADE
                |    1                                          If not corrected it will
                |    7                                          result in an impossible
                | CONFIRM:                                      output.
====>0          |                                         <====TYPE '0' TO REINPUT PAGE 2
                | TYPE PAGE 2:
====>920970985080070256336 17
                | RECEIVED FOR PAGE 2:
                | 92.0
                | 97.0
                | 98.5
                |   80
                |   70
                |   25
                |    6
                |    3
                |    3
                |    6
                |   17
                | CONFIRM:
====>1          |
                | TYPE PACE 3:
====>53047005023820203091850 35
                | RECEIVED FOR PAGE 3:
                | 53.0
                | 47.0
                |  5.0
                |  238
                |  202
                |  3.0
                |   91
                |   85
                |  3.5
                | CONFIRM:
====>1          |
                | TYPE PAGE 4:
====>160140035337
                | RECEIVED FOR PAGE 4:
                |  160
                |  140
                |  3.5
                |  337
                | CONFIRM:
====>1          |
```

TABLE XVII

Development Contract Summary

```
------------------------------------------------------------------------------

DP-3 *** DEVELOPMENT CONTRACT SUMMARY *** TEAM  3

                INCENTIVE PROVISIONS            INCENTIVE ACHIEVEMENTS

 INCENTIVE       WORST     BEST     MAX FEE      ACHIEVED
   AREA          VALUE     VALUE    ALLOWED       VALUE

DEV. COST       $ 53.0M  $ 47.0M     5.0%       $ 50.6M     $
FLT TST COMPL    238 WK   202 WK     3.0%        199 WK     $
RELIABILITY       85.0%     91.0%    3.5%         83.7%     $
ACCURACY         160YDS  140 YDS     3.5%       143 YDS     $

TOTALS                              15.0%                   $

TOTAL CONTRACT PRICE = $ 54.7M
```

```
          | COST + FACTORS ( NO)                           |
====>NO   |                                                |
          | DO YOU WISH TO CONTINUE, Y FOR YES,            |
          | N FOR NO                                       |
====>Y    |                                                | Y FOR ITERATION, N FOR STOP
          | TYPE DP NUMBER 3 OR 4                          |
====>3    |                                                |
          | THIS IS DP NUMBER ***** 3 *****                |
          | Y OR N                                         |
====>Y    |                                                |
          | TO CONFIRM, TYPE 1 FOR OK, 0 IF                |
          | CORRECTION MUST BE MADE.                       |
          | TEAM NUMBER                                    |
====>003  |                                                |
          | RECEIVED - TEAM NUMBER:     3                  |
          | CONFIRM:                                       |
====>1    |                                                |
          | NEW OR CHANGES TO INPUT DATA                   | CHANGES TO PAGES 2 AND 3 ONLY.
          |   RETYPE WHOLE PAGE CONTAINING ONE             |
          | 0 (ZERO) FOR NO - 1 (ONE) FOR YES              |
          | TYPE IN COMBINATION,4 0S AND 1S FOR            |
          | PAGES 1 to 4                                   |
====>0110 |                                                |
          | TYPE PAGE 2                                    |
====>880970985080070256336l7                               |
          | RECEIVED FOR PAGE 2:                           |
          | 88.0                                           |
          | 97.0                                           |
          | 98.5                                           |
          |   80                                           |
          |   70                                           |
          |   25                                           |
          |    6                                           |
          |    3                                           |
          |    3                                           |
          |    6                                           |
          |   17                                           |
          | CONFIRM:                                       |
====>1    |                                                |
          | TYPE PAGE 3:                                   |
====>54248805023820202587810 40                            |
          | RECEIVED FOR PAGE 3:                           |
          | 54.2                                           |
          | 48.8                                           |
          |  5.0                                           |
          |  238                                           |
          |  202                                           |
          |  2.5                                           |
          |   87                                           |
          |   81                                           |
          |  4.0                                           |
          | CONFIRM:                                       |
====>1    |                                                |
          |                                                |
```

TABLE XVIII

Page Data Strings

```
------------------------------------------------------------------------------------
10301       1       1       2      1.75    0.75    0.0
10302     88.00   97.00   98.50   80.00   70.00   25.00    6.00    3.00    3.00    6.00
          17.00
10303     54.20   48.80    5.00  238.00  202.00    2.50   87.00   81.00    4.00
10304    160.00  140.00    3.50  337.00

                                       CHECK CAREFULLY (This is a repeat
                                                   of the input data).
```

TABLE XIX

Development Contract Summary

```
---------------------------------------------------------------------------------

DP-3 *** DEVELOPMENT CONTRACT SUMMARY *** TEAM  3

                 INCENTIVE PROVISIONS          INCENTIVE ACHIEVEMENTS

 INCENTIVE      WORST     BEST    MAX FEE     ACHIEVED
   AREA         VALUE     VALUE   ALLOWED      VALUE

DEV. COST      $ 54.2M  $ 48.8M    5.0%       $ 50.2M    $
FLT TST COMPL   238 WK   202 WK    2.5%        199 WK    $
RELIABILITY      81.0%    87.0%    4.0%         83.7%    $
ACCURACY        160YDS   140 YDS   3.5%       145 YDS    $

TOTALS                            15.0%                  $

TOTAL CONTRACT PRICE = $ 56.2M
```

```
         COST + FACTORS ( NO)
====>NO
         DO YOU WISH TO CONTINUE, Y FOR YES,
         N FOR NO
====>N                                            'Y' FOR ITERATION
         R; T=0.73/1.28 18:02:42                  'N' FOR STOP
====>LOGOFF
```

```
                DP-4                              NOTES
       Repeat login procedure.

          R; T=0.01/0.01 17:58:49
====>LINK 0543P 191 192                           LINKTO CNP PROGRAM
          DASD 192 FORCED R/O; R/W BY 1681P
          R(00102); T=0.01/0.01 17:59:02
====>ACC 192 C
          C (192) R/O
          R; T=0.01/0.01 17:59:08
====>CNP                                          CALLS CNP
          EXECUTION BEGINS...
          TYPE DP NUMBER 3 OR 4
====>4
          THIS IS DP NUMBER ***** 4 *****
          Y OR N
====>Y
          TO CONFIRM, TYPE 1 FOR OK, 0 IF
          CORRECTION MUST BE MADE.
          TEAM NUMBER   XXX
====>003
          RECEIVED - TEAM NUMBER:    3
          CONFIRM:
====>1
          HAS CORRECT DP-3 DATA BEEN ENTERED
          _ Y,N
====>N                                            <====MUST ENTER N EACH TIME YOU LOGON.
          TYPE PAGE 1 DP-3 INFO
====>112017500750000
          CONFIRM:                                FIRST TIME THROUGH YOU MUST
====>1                                                  ENTER FINAL DP-3 DATA.
          TYPE PAGE 2 DP-3 INFO
====>88097098508007002563361 7                    CHECK EACH INPUT CAREFULLY
          CONFIRM:                                      BEFORE CONFIRMING
====>1
          TYPE PAGE 3 DP-3 INFO
====>54248805023820202587810 40
          CONFIRM:
====>1
          TYPE PAGE 4 DP-3 INFO
====>160140035337
          CONFIRM:
====>1
```

TABLE XX

Page Data Strings

DP-3										
PG-1	1	1	2	1.75	0.75	0.0				
PG-2	88.00 17.00	97.00	98.50	80.00	70.00	25.00	6.00	3.00	3.00	6.00
PG-3	54.20	48.80	5.00	238.00	202.00	2.50	87.00	81.00	4.00	
PG-4	160.00	140.00	3.50	337.00						

```
          | CONFIRM:                                  | DOUBLE CHECK OF CORRECT DP-3
====>1    |                                           |      ALL OUTPUT DEPENDS ON THIS.
          | NEW OR CHANGES TO INPUT DATA              | FIRST TIME YOU MUST
          |   RETYPE WHOLE PAGE CONTAINING ONE        |      ENTER ALL 1's (IN THIS
          | 0 (ZERO) FOR NO - 1 (ONE) FOR YES         |      EXAMPLE, ENTRY WAS '1110'
          | TYPE IN COMBINATION, 4 0S AND 1S FOR      |      BECAUSE PAGE-4 IS THE
          | PAGES 1 to 4                              |      SAME AS FOR DP-3.
====>1110 |                                           | THIS PRODUCES AN ERRONEOUS RESULT.
          | TYPE PAGE 1                               |      SEE BELOW.
====>112200075000
          | RECEIVED FOR PAGE 1:                      |
          |     1                                     |
          |     1                                     |
          |     2                                     |
          | 2.00                                      |
          | 0.75                                      |
          | 0.0                                       |
          | CONFIRM:                                  |
====>1    |                                           |
          | TYPE PAGE 2:                              |
====>8809709850800700276336l3
          | RECEIVED FOR PAGE 2:                      |
          | 88.0                                      |
          | 97.5                                      |
          | 98.5                                      |
          |    80                                     |
          |    70                                     |
          |    27                                     |
          |     6                                     |
          |     3                                     |
          |     3                                     |
          |     6                                     |
          |    13                                     |
          | CONFIRM:                                  |
====>1    |                                           |
          | TYPE PAGE 3:                              |
====>53247806018922501587810 40
          |                                           |
          | RECEIVED FOR PAGE 3:                      |
          | 53.2                                      |
          | 47.8                                      |
          |  6.0                                      |
          |  225                                      | (189) <====TEAM
          |  189                                      | (225)      REVERSED LIMITS AND
          |  1.5                                      | PUT IN 189 BEFORE 225.
          | *** INCENTIVE LIMITS WERE REVERSED ***    |
          |   87                                      | CNP MAKES THE CORRECTION
          |   81                                      |      AND GIVES A WARNING
          |  4.0                                      |
          | CONFIRM:                                  |
====>1    |                                           |
          | *** WARNING ***                           |
          | *** INCENTIVE LIMITS WERE REVERSED ***    |
          | ** DECISION SHEET MUST BE CORRECTED **    |
          |                                           |
          |                                           |
          |                                           |
          |                                           |
```

TABLE XXI

Page Data Strings

```
----------------------------------------------------------------------------------
10301     1      1      2      2.00   0.75   0.0
10302    88.00  97.00  98.50  80.00  70.00  25.00   6.00   3.00   3.00   6.00
         13.00
10303    53.20  47.80   6.00 225.00 189.00   1.50  87.00  81.00   4.00
10304     0.0    0.0    0.0    0.00                <====NOTE CNP FILLED IN
                                                     ALL 0'S FOR PAGE 4 .
```

TABLE XXII

Development Contract Achieved Value Summary

```
------------------------------------------------------------------------------

DP-4 *** DEVELOPMENT CONTRACT SUMMARY *** TEAM  3

                 INCENTIVE PROVISIONS           INCENTIVE ACHIEVEMENTS

 INCENTIVE       WORST     BEST    MAX FEE      ACHIEVED
   AREA          VALUE     VALUE   ALLOWED       VALUE

DEV. COST      $ 53.2M  $ 47.8M     6.0%       $ 49.0M    $
FLT TST COMPL   225 WK   189 WK     1.5%        186 WK    $
RELIABILITY      81.0%    87.0%     4.0%         82.0%    $
ACCURACY          OYDS    0 YDS     0.0%       149 YDS    $

TOTALS                             11.5%                  $
TOT MAX FEE ALLOWED DOES NOT EQUAL                          15.0%

TOTAL CONTRACT PRICE = $ 53.1M
```

```
              COST + FACTORS ( NO)                      NOTE  CNP WILL GIVE
====>NO                                                       A WARNING IF FEE DOES
              DO YOU WISH TO CONTINUE, Y FOR YES,             NOT EQUAL 15%.
              N FOR NO                                  IN FINAL BATCH PRINTOUT
                                                              THE PROGRAM WILL REALLOCATE
                                                              PERCENTAGE AMONG ALL VARIABLES.
====>Y                                                  'Y' FOR ITERATION TO GET
              TYPE DP NUMBER 3 OR 4                           CORRECT DATA IN.
====>4
              THIS IS DP NUMBER ***** 4 *****
              Y OR N
====>Y
              TO CONFIRM, TYPE 1 FOR OK, 0 IF
              CORRECTION MUST BE MADE.
              TEAM NUMBER   XXX
====>003
              RECEIVED - TEAM NUMBER:    3
              CONFIRM:
====>1
              HAS CORRECT DP-3 DATA BEEN ENTERED
              _ Y,N
====>Y                                                  BECAUSE DP-3 DATA IS ALREADY IN
              NEW OR INPUT CHANGES TO INPUT DATA
                RETYPE WHOLE PAGE CONTAINING ONE
              0 (ZERO) FOR NO - 1 (ONE) FOR YES
              TYPE IN COMBINATION,4 OS AND 1S FOR
              PAGES 1 to 4
====>1111
              TYPE PAGE 1
====>1122000750000
              RECEIVED FOR PAGE 1:
                  1
                  1
                  2
              2.00
              0.75
               0.0
              CONFIRM:
====>1
              TYPE PAGE 2:
====>88097098508007027633613
              RECEIVED FOR PAGE 2:
              88.0
              97.0
              98.5
                80
                70
                27
                 6
                 3
                 3
                 6
                13
              CONFIRM:
====>1
```

```
         | TYPE PAGE 3:
====>5324780602251890158781040
         | RECEIVED FOR PAGE 3:
         | 53.2
         | 47.8
         |  6.0
         |  225                                     | <====TEAM CORRECTED LIMITS
         |  189                                     |      ON DECISON SHEET.
         |  1.5
         |   87
         |   81
         |  4.0
         | CONFIRM:
====>1   |
         | TYPE PAGE 4:
====>160140035337
         | RECEIVED FOR PAGE 4:
         | 160
         | 140
         | 3.5
         | 337
         | CONFIRM:
====>1   |
```

TABLE XXIII

Development Contract Achieved Value Summary

```
--------------------------------------------------------------------------------

DP-4 *** DEVELOPMENT CONTRACT SUMMARY *** TEAM  3

                   INCENTIVE PROVISIONS          INCENTIVE ACHIEVEMENTS

 INCENTIVE      WORST      BEST     MAX FEE      ACHIEVED
   AREA         VALUE      VALUE    ALLOWED       VALUE

DEV. COST      $ 53.2M   $ 47.8M     6.0%       $ 49.0M    $
FLT TST COMPL   225 WK    189 WK     1.5%        186 WK    $
RELIABILITY      81.0%     87.0%     4.0%         82.0%    $
ACCURACY        160YDS   140 YDS     3.5%       149 YDS    $

TOTALS                              15.0%                  $

TOTAL CONTRACT PRICE = $ 54.3M
```

```
          COST + FACTORS ( NO)
====>NO
          DO YOU WISH TO CONTINUE, Y FOR YES,
          N FOR NO
====>N                                              'Y' FOR ITERATION
          R; T=0.73/1.28 18:02:42                   'N' FOR STOP
====>LOGOFF
```

APPENDIX B

DMS DATASHEETS

Team Number ___
DP-3

DEFENSE MANAGEMENT SIMULATION

DECISION SHEET 3

1. Of the total funds approved by the Secretary of Defense, the following distribution is made for procurement of the ZEBRA Weapon System:

a. Development Phase $___ ___.___ ___ Million [1-4]

b. Production Phase $___ ___.___ ___ Million [5-8]

2. The Development Contractor selected for ZEBRA System development is: (enter appropriate contractor number)

a. Midas Missile Corporation (contractor 1)
b. Apex Aerospace, Incorporated (contractor 2) ___ [9]

3. The basic guidance approach selected for the ZEBRA System is: (enter appropriate approach number)

a. Radar Inertial Guidance (G-1) (approach 1)
b. All-Inertial Guidance (G-2) (approach 2) ___ [10]

4. The guidance configuration selected for development is: (enter appropriate configuration number)

a. "A" as primary, "B" as parallel (configuration 1)
b. "A" alone (configuration 2)
c. "B" as primary, "A" as parallel (configuration 3)
d. "B" alone (configuration 4) ___ [11]

5. The development contract to be negotiated contains the following features:

a. Funds for the following activities. These funds are included in the total target development cost:

(1) Maintainability Engineering $___ ___.___ ___ Million [12-15]

(2) Value Engineering $___ ___.___ ___ Million [16-19]

(3) Parallel Guidance Development (enter 0 where not desired) $___ ___.___ ___ Million [20-23]

Team Number ___ DP-3

DECISION SHEET 3

b. The following minimum technical performance characteristics: (demonstrated at completion of flight tests)

(1) Motor Sub-system Reliability ___ ___.___ % [24-26]

(2) Airframe Sub-system Reliability ___ ___.___ % [27-29]

(3) Launcher/GSE Sub-system Reliability ___ ___.___ % [30-32]

(4) Fire Control Sub-system Impact Error ___ ___ ___ yards [33-35]

(5) Guidance Sub-system Impact Error ___ ___ ___ yards [36-38]

c. The following numbers of sub-system Qualification Tests:

(1) Motor Sub-system (20-40) ___ ___ [39-40]

(2) Airframe Sub-system (3-9) ___ [41]

(3) Launcher/GSE Sub-system (2-6) ___ [42]

(4) Fire Control Sub-system (2-4) ___ [43]

(5) Guidance Sub-system (3-9) ___ [44]

d. The following number of Flight Tests: (10-25) ___ ___ [45-46]

6. Decision Point ___ [76]

Team Number ___ ___ [77-78]

Card Number ___ [80]

Team Number ___ ___
DF-3

DECISION SHEET 3

7. The incentive provisions for this development contract (subject to negotiations) are as follows:

NOTE: If any area is not incentivized, enter the target value for both the maximum and minimum fee values, enter 0 for the maximum fee percentage.

a. Development Program Cost:

Maximum Cost (minimum fee value) $___ ___.___ Million [1-3]

Minimum Cost (maximum fee value) $___ ___.___ Million [4-6]

Maximum Fee Percentage ___ ___.___ % [7-9]

b. Schedule; Completion of Flight Tests:

Latest Week (minimum fee value) Week No. ___ ___ ___ [10-12]

Earliest Week (maximum fee value) Week No. ___ ___ ___ [13-15]

Maximum Fee Percentage ___ ___.___ % [16-18]

c. System Reliability:

Maximum Reliability (maximum fee value) ___ ___ % [19-20]

Minimum Reliability (minimum fee value) ___ ___ % [21-22]

Maximum Fee Percentage ___ ___.___ % [23-25]

DECISION SHEET 3

d. System Impact Error:

Maximum Error (poorest accuracy)
(minimum fee value) ___ ___ ___ yards [26-28]

Minimum Error (best accuracy)
(maximum fee value) ___ ___ ___ yards [29-31]

Maximum Fee Percentage ___ ___ ___ % [32-34]

8. Production Decisions:

a. The desired date for completion of deployment of lot TEN is: . Week No. ___ ___ ___ [44-46]

b. Fabrication of Block I missiles is to start after:

(1) Qualification Tests
[concurrent (fabrication 1)]
(2) Flight Tests
[fly-before-buy (fabrication 2)]

(1 or 2) ___ [47]

9. Class ___ ___ [73-74]

Decision Point ___ [76]

Team Number ___ ___ [77-78]

Card Number ___ [80]

Team Number ___
DP-4

DEFENSE MANAGEMENT SIMULATION

DECISION SHEET 4

1. The Development Contractor selected for ZEBRA System development is: (enter appropriate contractor number)

   a. Midas Missile Corporation (contractor 1)
   b. Apex Aerospace, Incorporated (contractor 2) ___ [1]

2. The basic guidance approach selected for the ZEBRA System is: (enter appropriate approach number)

   a. Radar Inertial Guidance (G-1) (approach 1)
   b. All-Inertial Guidance (G-2) (approach 2) ___ [2]

3. The final guidance configuration selected for development and production is: (enter appropriate configuration number)

   a. "A" configuration (configuration 2)
   b. "B" configuration (configuration 4) ___ [3]

4. The revised development contract contains the following features:

   a. Funds for the following activities:

      (If you do not desire to expend additional funds in these areas, enter the appropriate amounts expended at DP-3. If you do desire to expend additional funds in these areas, enter the total amounts you now desire.)

      (1) Maintainability Engineering $___.___ ___ Million [4-6]

      (2) Value Engineering $___.___ ___ Million [7-9]

   b. [For Monitor's Use Only] 0 0 0 [10-12]

Team Number ___
DP-4

DECISION SHEET 4

c. The following minimum technical performance characteristics:
(demonstrated at completion of flight tests)

(1) Motor Sub-system Reliability ___ ___.___ % [33-35]

(2) Airframe Sub-system Reliability ___ ___.___ % [36-38]

(3) Launcher/GSE Sub-system Reliability ___ ___.___ % [39-41]

(4) Fire Control Sub-system Impact Error ___ ___ ___ yards [42-44]

(5) Guidance Sub-system Impact Error ___ ___ ___ yards [45-47]

d. The following numbers of sub-system Qualification Tests:

(1) Motor Sub-system (20-40) ___ ___ [48-49]

(2) Airframe Sub-system (3-9) ___ [50]

(3) Launcher/GSE Sub-system (2-6) ___ [51]

(4) Fire Control Sub-system (2-4) ___ [52]

(5) Guidance Sub-system (3-9) ___ [53]

e. The following number of Flight Tests: (10-25) ___ ___ [54-55]

5. Decision Point ___ [76]

Team Number ___ ___ [ ]

Card Number ___ [ ]

Team Number ___ DP-4

DECISION SHEET 4

6. The incentive provisions for this development contract (subject to negotiations) are as follows:

NOTE: If any area is not incentivized, enter the target value for both the maximum and minimum fee values, enter 0 for the maximum fee percentage.

a. Development Program Cost:

Maximum Cost (minimum fee value) $___ ___.___ Million [1-3]

Minimum Cost (maximum fee value) $___ ___.___ Million [4-6]

Maximum Fee Percentage ___ ___.___ % [7-9]

b. Schedule; Completion of Flight Tests:

Latest Week (minimum fee value) Week No. ___ ___ ___ [10-12]

Earliest Week (maximum fee value) Week No. ___ ___ ___ [13-15]

Maximum Fee Percentage ___ ___.___ % [16-18]

c. System Reliability:

Maximum Reliability (maximum fee value) ___ ___ % [19-20]

Minimum Reliability (minimum fee value) ___ ___ % [21-22]

Maximum Fee Percentage ___ ___.___ % [23-25]

DECISION SHEET 4

d. System Impact Error:

Maximum Error (poorest accuracy)
(minimum fee value) ___ ___ ___ yards [26-28]

Minimum Error (best accuracy)
(maximum fee value) ___ ___ ___ yards [29-31]

Maximum Fee Percentage ___ ___ ___ % [32-34]

7. Production Decisions:

a. The Development Contractor is to procure the following number of lots of long-lead time components in preparation for Block I production:

___ ___ [44-45]

b. The earliest Lot TEN deployment date for which fixed price proposals are desired is:

Week No. ___ ___ ___ [46-48]

c. Fabrication of Block I missiles is to start after:

(1) Qualification Tests
[concurrent (fabrication 1)]
(2) Flight Tests
[fly-before-buy (fabrication 2)]

(1 or 2) ___ [49]

8. Class ___ ___ [73-74]

Decision Point ___ [76]

Team Number ___ ___ [77-78]

Card Number ___ [80]

APPENDIX C

LINKPROJ EXEC

```
*THIS EXEC FACILITATES STUDENT LINK TO THE 0276P DISK FOR EXECUTION OF
*THE PROJECT MANAGER SIMULATION "PROJMNG FORTRAN".
&CONTROL OFF
-AGAIN
CP LINK 0276P 191 192 RR
ACCESS 192 C
CP SET PF09 IMM REL 192 (DET) #CP SET PF09
CLRSCRN
EXEC RECORD ON
&TYPE * NOTE!: YOU ARE NOW LINKED TO PROJMNG ON YOUR 192 DISK, MODE C
&TYPE * NOTE!: PRESS PF09 TO BREAK THE LINK
EXEC PROJMNG
REL 192 (DET)
CP SMSG QACNT KLINE
&BEGTYPE
                       ANOTHER RUN, 'Y'/'N'?
&END
&READ ARGS
&IF .&1 EQ . &SKIP
&IF &1 EQ Y &GOTO -AGAIN
EXEC RECORD OFF
-END
```

APPENDIX D

PROJMNG EXEC

```
*"PROJMNG EXEC" PROVIDES ENTRY CHECKS FOR THE MAIN PROGRAM "PROJMNG".
*AMONG THE CHECKS ARE THOSE HIGHLIGHTED BY COMMENTS BELOW. THEY FACIL-
*ITATE "TEAM" STORED FILE PROCESSING.
*****..................................................................
&CONTROL OFF
SET BLIP *
SET CMSTYPE HT
GLOBAL TXTLIB FORTMOD2 MOD2EEH IMSLSP NONIMSL CMSLIB
&FLAG = 0
&ONE  = 21
FILEDEF 5 TERMINAL (PERM
FILEDEF 6 TERMINAL (PERM
FILEDEF 13 DISK DATAINST DISKNUM A(PERM
FILEDEF 17 DISK FILE FT17F001 C(PERM
FILEDEF 18 DISK FILE FT18F001 C(PERM
FILEDEF 19 DISK FILE FT19F001 C(PERM
FILEDEF 20 DISK FILE FT20F001 C(PERM
FILEDEF 23 DISK FILE FT23F001 C(PERM
*FILE DATAINST--IDENTIFIES THE DISK AS BELONGING TO A MONITOR/INSTRUCTOR
*FILES 17,18,19 & 20 -- ARE DATA FOR ACHIEVED VALUE DETERMINATION IN
*               EXECUTING THE PROGRAM'S SIMULATED INTERACTION WITH A
*               CONTRACTOR..............................................
******..................................................................
SET CMSTYPE RT
*    (THE FOLLOWING COPYFILES ARE PERFORMED TO PERMIT FORTRAN PROGRAM
*                                    EXECUTION TO CHANGE THE TEAM FILES)
USERID
*                          (USERID VARIABLES ARE USED AS SECURITY CHECKS)
&READ VARS &DUMMY &USERID
&USER = 0
*                 (CHECK FOR THE PRESENCE OF AN INSTRUCTOR PECULIAR FILE)
STATE DATAINST DISKNUM A
&RET = &RETCODE
&IF &RET NE 0 &GOTO -STUDENT
*                      (CHECK FOR A "P" IN THE FIRST LINE FIFTH POSITION)
FINDSTAK DATAINST DISKNUM A 5 P
&IF &RETCODE NE 0 &GOTO -STUDENT
&READ VARS &USER
&IF &USER NE &USERID &GOTO -STUDENT
&BEGTYPE
```

```
             PRESS <<ENTER>> WHEN YOU ARE READY TO CONTINUE.
&END
&READ
CLRSCRN
&BEGTYPE
******.....EXECUTION IS IN PROGRESS. WAIT. DO NOT PRESS ENTER.******
&END
FILEDEF 11 DISK DATAINST DISKNUM A
*                           (ESTABLISH THE INSTRUCTOR UNIQUE FLAG OF 21)
&FLAG = TEAM21
STATE DATAFILE TEAM21 A
&IF &RETCODE NE 0 COPYFILE FILE DATAFILE C DATAFILE TEAM21 A (REPLACE
&NM = 0
*                                             (CHECK FOR DATAFILES 1-20)
-NEXTIN
&NM = &NM + 1
&CHART = TEAM&NM
&IF &NM LT 10 &CHART = TEAM0&NM
STATE DATAFILE &CHART C
&IF &RETCODE EQ 0 COPYFILE DATAFILE &CHART C = = A (REPLACE
&IF &NM LT 20 &GOTO -NEXTIN
FILEDEF 9 DISK DATAFILE TEAM21
&BEGTYPE
             PRESS <<ENTER>> WHEN YOU ARE READY TO CONTINUE.
&END
&READ
-PROCES
CLRSCRN
**********************************************************************
*                                                                    *
*  THE FOLLOWING LINES PROVIDE A ROUTINE TO PURGE TEAM DATAFILES     *
*  FROM THE DISK ON WHICH THE USER HAS 'A' ACCESSED. THE DESIGN OF   *
*  THIS FEATURE IS TO PURGE ALL TEAM FILES FROM THE DISK ONTO WHICH  *
*  TEAMFILES ARE SENT BY THIS EXEC ROUTINE. THIS OPERATION WILL BE   *
*  PARTICULARLY BENEFICIAL IN HELPING MONITORS TO INITIALIZE THE     *
*  DISK FOR A NEW GAME.                                              *
*  THIS IS ACCOMPLISHED BY ANSWERING 'Y', AFFIRMATIVE, TO PURGE TEAM *
*  DATAFILES WHILE LOGGED ONTO THE PROJMNG MOUNTED FILE.             *
*                                                                    *
**********************************************************************
```

```
&BEGTYPE
                         ******WARNING******
       *****ANSWERING 'Y'(=YES) TO THE FOLLOWING QUERY WILL*****
            *****PURGE ALL EXISTING STUDENT DATAFILES******
                 *****ON THE DISK YOU ARE USING.*****

    DO YOU WISH TO INITIALIZE ALL TEAM FILES FOR A NEW DMS EXERCISE?
                   'PURGE' =PURGE ALL TEAM DATAFILES
                   'N'=PROCEED WITH NORMAL OPERATION
                   'E'=STOP

&END
CMS MAKEBUF
&READ VARS &N
CMS DROPBUF
&IF .&N EQ . &GOTO -PROEX
&IF &N EQ PURGE &GOTO -REDUM
&IF &N EQ E &GOTO -END
&GOTO -PROGRUN
-PROEX
&BEGTYPE
               DO YOU WISH TO CONTINUE; Y/N ?
&END
&READ VARS &NOGO
&IF .&NOGO EQ . &GOTO -INSTEND
&IF &NOGO EQ Y &GOTO -PROCES
&GOTO -INSTEND
-REDUM
STATE DATAFILE * A
&IF &RETCODE NE 0 &GOTO -END
ERASE DATAFILE * A
ERASE DATACODE * A
&IF &RETCODE NE 0 &GOTO -END
&GOTO -REDUM
-STUDENT
*                                                  (GET THE TEAM NUMBER)
STATE DATAFILE * A
&RET = &RETCODE
&IF &RET NE 0 &GOTO -NEWNUM
*                    (CHECK FOR A "3" IN THE FIRST LINE FOURTH POSITION)
```

```
FINDSTAK DATAFILE * A
&READ VARS &ONE &TWO &THREE
&IF &ONE EQ 0 &GOTO -NEWNUM
&GOTO -TEAMNUM
-NEWNUM
&BEGTYPE
                       WHAT IS YOUR TEAM NUMBER?
&END
CMS MAKEBUF
&READ VARS &ONE
CMS DROPBUF
&IF .&ONE EQ . &GOTO -EXITTM
&IF &ONE GE 21 &GOTO -EXITTM
&IF &ONE LE 0 &GOTO -EXITTM
&IF &ONE GT 0 &GOTO -TEAMNUM
-EXITTM
&BEGTYPE
               DO YOU WISH TO CONTINUE; Y/N ?
&END
&READ VARS &NOGO
&IF .&NOGO EQ . &GOTO -INSTEND
&IF &NOGO EQ Y &GOTO -NEWNUM
&GOTO -INSTEND
-TEAMNUM
&1 = &ONE
&CHAR = TEAM&1
&IF &1 LT 10 &CHAR = TEAM0&1
*                                (ESTABLISH THE STUDENT'S WORKING FILES)
FILEDEF 9 DISK DATAFILE &CHAR *(PERM
&IF &FLAG NE TEAM21 FILEDEF 11 DISK DATACODE &CHAR *(PERM
*     (CHECK THE INSTRUCTOR FILES FOR A PREVIOUSLY STORED TEAM DATABASE)
-TEAMFILE
STATE DATAFILE &CHAR C
&IF &RETCODE NE 0 &GOTO -ADISKM
FINDSTAK DATAFILE &CHAR C
&READ VARS &THREE
&IF .&THREE EQ . &GOTO -ADISKN
&IF &THREE EQ 0 &GOTO -ADISKN
COPYFILE DATAFILE &CHAR C = = A (REPLACE)
-ADISKN
```

```
STATE DATAFILE &CHAR A
&IF &RETCODE EQ 0  &GOTO -CODE
COPYFILE FILE DATAFILE C DATAFILE &CHAR A (REPLACE)
-CODE
STATE DATACODE &CHAR C
&IF &RETCODE NE 0 &GOTO -ADISKC
FINDSTAK DATACODE &CHAR C
&READ VARS &THREE
&IF .&THREE EQ . &GOTO -ADISKC
&IF &THREE EQ 0 &GOTO -ADISKC
COPYFILE DATAFILE &CHAR C = = A (REPLACE)
-ADISKC
STATE DATACODE &CHAR A
&IF &RETCODE EQ 0 &GOTO -PROGRUN
COPYFILE FILE DATACODE C DATACODE &CHAR A (REPLACE)
SET CMSTYPE RT
-PROGRUN
&BEGTYPE
                     WAIT FOR "EXECUTION TO BEGIN".
&END
SET BLIP OFF
&STACK &ONE
&STACK &FLAG
**************************************************************************
LOAD PROJMNG (START
**************************************************************************
&IF &READFLAG EQ STACK DESBUF
SET BLIP *
* (THE FORTRAN PROGRAM MANIPULATION OF FILES PREVENTS GENERATION OF NEW
* DATA FILES DURING THE PROGRAM EXECUTION. THE FOLLOWING ROUTINE PERMITS
* THE OPERATOR TO GENERATE ATLEAST ONE NEW DATABASE FILE BY PROGRAM RE-
* DEFINITION OF FILEDEFS AS APPLIED TO THE THREE DUMMY FILES GENERATED
* AT THE TOP OF THIS "EXEC".)
* (IF THE OPERATOR IS NOT AN INSTRUCTOR, BYPASS DATAFILE PRINTOUT
*                                                           CAPABILITIES)
&IF &USERID NE &USER &GOTO -NOTINS
*                                                    (GET THE TEAM NUMBER)
&BEGTYPE
      DO YOU WISH TO RECEIVE A LISTING OF TEAMS ON THE CLASS DISK,
      Y=YES?
```

```
&END
&READ VARS &ANS
&IF .&ANS NE . &GOTO -INSTEND
&IF &ANS NE Y &GOTO -INSTEND
*                               (INITIALIZE THE DATAFILE INCREMENT TO 0)
&NM = 0
*                                             (CHECK FOR DATAFILES 1-20)
CP SPOOL PUNCH CONT TO &USER
-NEXTNM
&NM = &NM + 1
&CHART = TEAM&NM
&IF &NM LT 10 &CHART = TEAMO&NM
 PUNCH DATAFILE &CHART C
&IF &NM LT 20 &GOTO -NEXTNM
CP SPOOL PUNCH CLOSE NOCONT
CP SPOOL PUNCH OFF
READCARD INST FILE A
&GOTO -INSTEND
-NOTINS
*                (DATA DUMP THE TEAM'S DATABASE FILES TO THE INSTRUCTOR)
CP SPOOL PUNCH CL X CONT TO 0276P
PUNCH DATAFILE &CHAR A0
&PUNCH &USERID
CP SPOOL PUNCH CLOSE NOCONT
CP SPOOL PUNCH OFF
CP SPOOL PUNCH CL X CONT TO 0276P
PUNCH DATACODE &CHAR A0
&PUNCH &USERID
CP SPOOL PUNCH CLOSE NOCONT
CP SPOOL PUNCH CL A
CP SPOOL PUNCH OFF
-INSTEND
SET CMSTYPE HT
ERASE LOAD MAP A
SET CMSTYPE RT
SET BLIP OFF
-END
```

APPENDIX E

PROFILE EXEC FOR PROJMNG

```
&TRACE
* The following warning alerts operators logging onto the disk
* that they should access by "ACC (NOPROF" and not permit the
* system to run the Profile Exec since it has a logoff command
* on its bottom line. Normal Profile type functions are performed
* by executing a separate command of "MYPROF" which performs the
* routines in MYPROF EXEC.
&BEGTYPE
   ******WARNING******  <<ACC (NOPROF>>,<<MYPROF>> VICE <<PROFILE>>
&END
CP SPOOL RDR CL X
READ *
READ *
READ *
READ *
READ *
READ *
READ *
READ *
CP SPOOL RDR CL *
CP LOGOFF
```

APPENDIX F

PROJMNG STUDENT INSTRUCTIONS

A. INTRODUCTION

This pamphlet contains general instructions for student operation of the program "PROJMNG FORTRAN" which is designed as a companion to the Industrial College of the Armed Forces' "DEFENSE MANAGEMENT SIMULATION" exercise.

The program functions as a computerized analysis tool assisting a project manager in evaluating his assessment of the goals and incentives for his contract to procure a new missile system. It does not provide answers to the simulation's main batch computer program.

Significant effort has been placed in making the program as easy to operate as possible. The user-friendly design permits the simulation's participants to correct, to rerun, to submit reports to the monitor, and to exit the game with the minimum of difficulty.

Additionally, consideration has been given to protecting the individual team's results. A generic security system guards access to the team's files to prevent unintentional destruction of the database, or database plagiarism. Proper entry of the team security code enables the team to access its stored database and permits submission of a team proposal to the class monitor. Attempts to circumvent these measures are counted and recorded in the database. After five erroneous entries of the security code the team's data

files are not accessible by the students. The monitor must be contacted to reset the counter. In order to document the access and attempts to access a team database, the routine PROJMNG EXEC transmits the database to the monitor's computer disk with the USERID and USERNAME from the disk running PROJMNG EXEC.

In the event you dump (exit) out of the program, other files may appear on your disk such as PROJECT MODULE, LOAD MAP, FILE DATAFILE, and FILE DATACODE. Please erase these extra files. The FILEDEFs in Table XXIV will be generated DO NOT ALLOW FILES ON YOUR DISK WHICH ARE NOT PART OF PROJMNG TO USE THESE FILE NAMES.

TABLE XXIV

PROJMNG FILEDEFS

```
1. FILE  5    TERMINAL   READ
2. FILE  6    TERMINAL   PRINT
3. FILE  9    DISK       DATAFILE TEAMxx A
4. FILE 11    DISK       DATACODE TEAMXX A
5. FILE 17    DISK       FT17F001 C   (DATAFILES)
6. FILE 18    DISK       FT18F001 C   (DATAFILES)
7. FILE 19    DISK       FT19F001 C   (DATAFILES)
8. FILE 20    DISK       FT20F001 C   (DATAFILES)
```

during the execution of PROJMNG EXEC .

The following sequence of operations demonstrates the procedures involved in the operation of "PROJMNG FORTRAN". Several files are involved in the game operation. They are listed in order to enable the participant to manage his student disk. Problems with file definition conflict should be considered. Do not leave non-game files with the same file name on your disk.

flagging of the program for access to the next DP level. Please leave this file on your disk during your use of the game.

5. DATACODE TEAMxx contains your team's security code and number of times it has been incorrectly entered. Once you terminate a run of the simulation, the file is sent to the monitor who uses it as his copy. On the monitor's PROJMNG disk, it provides the primary code file for the team. The monitor's copy is the primary copy in order to preclude program abort if DATACODE cannot be found on your disk. DO NOT TAMPER WITH THESE FILES or difficulty will occur in the program's correlation of your team's security code.

B. INSTRUCTIONS

PROJMNG FORTRAN is available on the Naval Postgraduate School IBM-3033 computer in the 'W. R. CHURCH COMPUTER CENTER'.

The text which follows is a tutorial demonstrating the features of the PROJMNG package of programs and data files. These features are demonstrated as they would occur on the IBM-3033 using an IBM 3278 terminal. Follow the procedures as they appear in the text. Use your own team number and team security code.

In the following examples, please perform the operations highlighted in the left margin by ====>.

The three column divisions below are arranged with the operator inputs in the left column. Computer responses and the headings for both explanation and for *****EXAMPLEs***** are in the center column. The right column contains remarks to aid in understanding the procedures being executed.

====>01kline	ENTER YOUR NEW CODE.	ized to zeros. PROJMNG determines if you have a code of 8 zeros and requires a code input to be made. Select a code of up to 8 characters and enter it. If you already have a code, the program will ask you to enter it for a comparative validation. The program will now proceed to check the database for previous information. If it finds nothing in the files, it will sequence through each item from the participant's decision point data sheets asking for the inputs. At the end of each page it will check the entries against the input requirements and ask for corrections as necessary.

```
OPERATOR       PROGRAM                                                   R E M A R K S

====>logon
====>linkproj  BEGIN RECORDING OF TERMINAL SESSION
               * NOTE!: YOU ARE NOW LINKED TO PROJMNG ON YOUR 192 DISK
               MODE C
               * NOTE!: PRESS PF09 TO BREAK THE LINK
               FILE 'DATAINST DISKNUM A' NOT FOUND.                      PROJMNG EXEC checks to see if you are an
               FILE 'DATAFILE * A' NOT FOUND.                                    instructor(monitor).
               WHAT IS YOUR TEAM NUMBER?                                 If not, and if it finds no TEAMxx data files,
                                                                                 it will create the files needed.
====>1                                                                   You should enter your team's number here.
                                                                                 Wait for completion of this process.
               FILE 'DATAFILE TEAM01 C' NOT FOUND.                       Determines if a DATAFILE TEAMxx has already
               FILE 'DATAFILE TEAM01 A' NOT FOUND.                               been submitted to the monitor.
               FILE 'DATACODE TEAM01 C' NOT FOUND.                       PROJMNG EXEC  creates all of its needed
               FILE 'DATACODE TEAM01 A' NOT FOUND.                               files on your 'A' disk.
               PRESS <<ENTER> WHEN YOU ARE READY TO CONTINUE

                              WAIT FOR "EXECUTION TO BEGIN".             There may be a pause of several minutes
               EXECUTION BEGINS...
                                                                                 in completing this process.
                                                                         DO NOT PANIC.
               ***********************************************************
                        TO TERMINATE THE PROGRAM AT ANY POINT,
                    TYPE "E"=EXIT IN RESPONSE TO ANY "YES/NO" QUERY.
               ***********************************************************
               ***CAUTION***IN RESPONSE TO A QUERY, PRESSING <<ENTER>>
                    WITHOUT PROVIDING DATA WILL DUMP THE PROGRAM.
               ***********************************************************

               ****YOU ARE ENCOURAGED TO CHANGE YOUR SECURITY CODE.***  All security codes have been initial-
```

```
             |.................... P  A  G  E  -  1..................|
             |  ______________________________________________________ |
             | |                                                      ||
             | | *****INPUT PAGE 1 DP-3 INFO*****.                    ||
====>1       | | CONTRACTOR NUMBER,(1 OR 2):?                         ||
====>1       | | GUIDANCE APPROACH,(1 OR 2):?                         ||
             | | GUIDANCE CONFIGURATION,(1-4;NOT 1 OR 3 AFTER DP-3?   ||Parallel guidance selections 1 or 3
====>1       | | MAINTAINABILITY ENGINEERING FUNDS IN $M:?            ||          will be rejected after DP-3.
====>1.75    | | VALUE ENGINEERING FUNDS IN $M:?                      ||
====>.75     | | PARALLEL GUIDANCE FUNDS IN $M.?                      ||Query for parallel guidance funds.
====>.5      | |                                                      ||If guidance configuration selected is 2
             | |______________________________________________________||          or 4, or after DP-3, parallel
             |...........D  A  T  A    C  H  E  C  K  S.............|          guidance funds will not be
             |                                                      |          queried. They are zeroed.
             |                                                      |
             |                                                      |          Checks of data values will occur
             |                                                      |          at the end of each page.
             |                                                      |          Necessary corrections will be
             |                                                      |          queried.
             |                                                      |Sample *****WARNINGS***** of data errors
             |                                                      |          are demonstrated below.
             |                                                      |
```

```
                 ....................P A G E - 2..................
                  ___________________________________________
                 |                                           |
                 | *****INPUT PAGE 2 DP-3 INFO.*****         |
====>90          | MOTOR RELIABILITY?                        |
====>97          | AIRFRAME RELIABILITY?                     |
====>98.5        | LAUNCHER/GSE RELIABILITY?                 |
====>80          | FIRE CONTROL ACCURACY?                    |
====>70          | GUIDANCE ACCURACY?                        |
====>25          | MOTOR QUAL TESTS:20-40?                   |
====>6           | AIRFRAME QUAL TESTS:3-9?                  |
====>3           | LAUNCHER/GSE QUAL TESTS:2-6?              |
====>3           | FIRE CONTROL QUAL TESTS:2-4?              |
====>1           | GUIDANCE SYSTEM QUAL TESTS:3-9?           |
====>7           | FLIGHT TESTS:10-25?                       |
                 |___________________________________________|

                 ...........D A T A   C H E C K S.............
                 ...........C O R R E C T I O N S.............

                           *****WARNING*****
                 GUIDANCE QUALIFICATION TESTS MUST BE GREATER THAN
                 3 AND LESS THAN 9. YOUR QUAL TESTS= 1.
                 PREVIOUS VALUE 1.00, INPUT A VALUE?
```

The two warnings below are samples of the
result of parameter checks at the
end of this data sheet page.

```
====>6
                        *****WARNING*****
               FLIGHT TESTS MUST BE GREATER THAN 10 AND LESS THAN 25.
               YOUR FLIGHT TESTS=7. previous value 7.00, INPUT A VALUE?
====>17
```

```
             ....................P  A  G  E  -  3....................
             ┌──────────────────────────────────────────────────┐
             │ *****INPUT PAGE 3 DP-3 INFO.*****                │
             │ MAXIMUM COST IN $M?                              │
====>53      │ MINIMUM COST IN $M?                              │
====>47      │ INCENTIVE FEE % FOR DEVELOPMENT COST?            │
====>5       │ LATEST WEEK FOR FLIGHT TEST COMPLETION?          │
====>238     │ EARLIEST WEEK FOR FLIGHT TEST COMPLETION?        │
====>202     │ INCENTIVE FEE % FOR TEST COMPLETION?             │
====>3       │ MAXIMUM RELIABILITY % EXPECTED?                  │
====>91      │ MINIMUM RELIABILITY % EXPECTED?                  │
====>92      │ INCENTIVE FEE % FOR RELIABILITY?                 │
====>3.5     │                                                  │
             └──────────────────────────────────────────────────┘
             ...........D  A  T  A    C  H  E  C  K  S............
             ...........C  O  R  R  E  C  T  I  O  N  S............

                         *****WARNING*****
             MAXIMUM SYSTEM RELIABILITY IS LESS THAN
             MINIMUM RELIABILITY: MAX = 91. ; MIN = 92.
             FOR MAXIMUM RELIABILITY,
             PREVIOUS VALUE 91.00, INPUT a VALUE.
====>91

             FOR MINIMUM RELIABILITY,
```

Following are additional examples of correction check warnings.

The program automatically queries pairs of values which are for the same data element.

```
          |..................P A G E - 4..................|
          | ______________________________________________ |
          ||                                              ||
          || *****INPUT PAGE 4 DP-3 INFO.*****            ||
          || MAXIMUM IMPACT ERROR EXPECTED?               ||
====>160  || MINIMUM IMPACT ERROR EXPECTED?               ||
====>140  || INCENTIVE FEE % FOR IMPACT ACCURACY?         ||
====>3.5  || INPUT DESIRED WEEK FOR COMPLETION OF         ||
          || DEPLOYMENT?                                  ||
====>337  ||______________________________________________||
          |...........D A T A   C H E C K S...........|
          |...........C O R R E C T I O N S...........|
```

```
             |FOR  92.00, INPUT a VALUE.
====>85      |
```

```
                  DATABASE PRINTOUT as SELECTION MENU
------------------------------------------------------------------------
NO.*ITEM******CURRENT VALUE @ DP-3    NO.*ITEM***********CURRENT VALUE
 1. CONTRACTOR            :  1        16. GUIDANCE QUAL TESTS   :  6.00
 2. GUIDANCE APPROACH     :  1        17. FLIGHT TESTS          : 17.00
 3. GUIDANCE CONFIGURATION:  1        18. MAXIMUM COST        M$: 53.00
 4. MAINTAINABILITY ENG M$:  1.75     19. MINIMUM COST        M$: 47.00
 5. VALUE ENG           M$:  0.75     20. MAX COST INCENTIVE   %:  5.00
 6. PARALLEL DEVELOP    M$:  0.5      21. LATEST WEEK           :238.00
 7. MOTOR RELIABILITY    %: 90.00     22. EARLIEST WEEK         :202.00
 8. AIRFRAME RELIABILITY %: 97.00     23. MAX DELIVER INCENTIVE%:  3.00
 9. LAUNCHER RELIABILITY %: 98.50     24. MAXIMUM RELIABILITY  %: 91.00
10. FIRE CONTROL ERROR YDS: 80.00     25. MINIMUM RELIABILITY  %: 85.00
11. GUIDANCE ERROR     YDS: 70.00     26. MAX RELIABIL INCENTIV%:  3.50
12. MOTOR QUAL TESTS      : 25.00     27. MAXIMUM ERROR      YDS:160.00
13. AIRFRAME QUAL TESTS   :  6.00     28. MINIMUM ERROR      YDS:140.00
14. LAUNCHER QUAL TESTS   :  3.00     29. MAX ERROR INCENTIVE  %:  3.50
15. FIRE CONTROL QUAL TSTS:  3.00     30. WEEK FOR LOT 10       :337.00
                  *****0. NONE*********31. INPUT BY PAGE
             SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3
```

Check each display carefully for erroneous inputs. The results computed are based on the values appearing in this table.

```
====>31
```

Change by page

```
*****TYPE IN A LIST OF PAGES YOU DESIRE TO CHANGE;*****
EX:1234 WILL ASK FOR NEW DATA ON ALL PAGES; OR, 23 WILL
ASK FOR NEW DATA ON PAGES 2 AND 3.
                 *****WARNING*****
YOU MUST ENTER AT LEAST ONE DIGIT EVEN IF IT IS A '0'.
```

```
====>12
```

Changing pages 1 and 2

```
                 .........R E E N T E R _ P A G E - 1........

                   *****INPUT PAGE 1 DP-3 INFO*****.
====>1             CONTRACTOR NUMBER,(1 OR 2):?
====>1             GUIDANCE APPROACH,(1 OR 2):?
                   GUIDANCE CONFIGURATION,(1-4;NOT 1 OR 3
                   AFTER DP-3?
====>2
====>1.75          MAINTAINABILITY ENGINEERING FUNDS IN M$:?
====>.75           VALUE ENGINEERING FUNDS IN M$:?

                 ...........D A T A   C H E C K S.............
                 ............C O R R E C T I O N S............

                              *****WARNING*****
                 PARALLEL DEV FUNDS OF $ 0.50 M
                 ARE NOT REQUIRED AND HAVE BEEN RESET TO $0.0

                 DO YOU WISH TO CHANGE GUIDANCE CONFIGURATION:
                 "Y"=YES OR "N"=NO?
====>N
```

Parallel guidance selections 1 or 3 will be rejected after DP-3.

NOTE No query for parallel guidance funds. If guidance configuration selected is 2 or 4, or after DP-3, parallel guidance funds will not be queried. They are zeroed.

Checks of data values will occur at the end of each page.

Necessary corrections will be queried. Sample *****WARNINGS***** of data errors are demonstrated below.

The value of parallel development funds at $0.5 million is not an accetable option.

The program now cues the operator to the problem by asking for a correction in the guidance configation which had prevented the desired change in parallel development funds.

```
                 .........R  E  E  N  T  E  R  _ P  A  G  E - 2........

                   *****INPUT PAGE 2 DP-3 INFO.*****
                   MOTOR RELIABILITY?
====>92            AIRFRAME RELIABILITY?
====>97            LAUNCHER/GSE RELIABILITY?
====>98.5          FIRE CONTROL ACCURACY?
====>80            GUIDANCE ACCURACY?
====>70            MOTOR QUAL TESTS?
====>25            AIRFRAME QUAL TESTS?
====>6             LAUNCHER/GSE QUAL TESTS?
====>3             FIRE CONTROL QUAL TESTS?
====>3             GUIDANCE SYSTEM QUAL TESTS?
====>6             FLIGHT TESTS?
====>17

                 ...........D  A  T  A    C  H  E  C  K  S............ Checks of data values will occur at the
                                                                                 end of each page.
                 ...........C  O  R  R  E  C  T  I  O  N  S........... Necessary corrections will be queried.
                                                                        Sample *****WARNINGS***** of data errors
                                                                                 are demonstrated below.
```

```
                              SELECTION MENU
 -------------------------------------------------------------------------
  NO.*ITEM******CURRENT VALUE @ DP-3     NO.*ITEM************CURRENT VALUE
  1.  CONTRACTOR            :  1         16. GUIDANCE QUAL TESTS    :  6.00
  2.  GUIDANCE APPROACH     :  1         17. FLIGHT TESTS           : 17.00
  3.  GUIDANCE CONFIGURATION:  2         18. MAXIMUM COST         M$: 53.00
  4.  MAINTAINABILITY ENG M$:  1.75      19. MINIMUM COST         M$: 47.00
  5.  VALUE ENG           M$:  0.75      20. MAX COST INCENTIVE    %:  5.00
  6.  PARALLEL DEVELOP    M$:  0.0       21. LATEST WEEK            :238.00
  7.  MOTOR RELIABILITY    %: 92.00      22. EARLIEST WEEK          :202.00
  8.  AIRFRAME RELIABILITY %: 97.00      23. MAX DELIVER INCENTIVE%:  3.00
  9.  LAUNCHER RELIABILITY %: 98.50      24. MAXIMUM RELIABILITY   %: 91.00
 10.  FIRE CONTROL ERROR YDS: 80.00      25. MINIMUM RELIABILITY   %: 85.00
 11.  GUIDANCE ERROR     YDS: 70.00      26. MAX RELIABIL INCENTIV%:  3.50
 12.  MOTOR QUAL TESTS      : 25.00      27. MAXIMUM ERROR      YDS:160.00
 13.  AIRFRAME QUAL TESTS   :  6.00      28. MINIMUM ERROR      YDS:140.00
 14.  LAUNCHER QUAL TESTS   :  3.00      29. MAX ERROR INCENTIVE   %:  3.50
 15.  FIRE CONTROL QUAL TSTS:  3.00      30. WEEK FOR LOT 10        :337.00
                *****0. NONE**********31. CHANGE BY PAGE*****
             SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3.

====>0
```

Zero from the Selection Menu sends the program to the Main Menu.

```
                                MAIN MENU

 ---------------------------------------------------------------

 DO YOU WISH TO:

 1. RECEIVE THE TABLE OF ACHIEVED VALUES.
                  *****WARNING*****AFTER THE TENTH DP- 3 RUN
                  AN ADDITIONAL DEVELOPMENT COST OF $100,000
                  WILL BE INCURRED FOR EACH ADDITIONAL RUN.
                  THIS IS RUN 1 OF DP- 3.
 2. INPUT SELECTION MENU
 3. SUBMIT A CONTRACT PROPOSAL
 4. EXIT
```

```
====>1
```

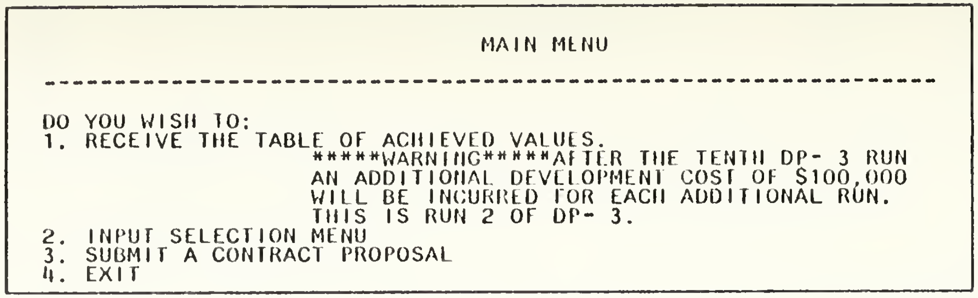
'1' tells the program to proceed with a check of the total fee percentage. It insures the total fee percentage, 15% of the development contract, is allocated. If the fees do not total 15%, the program queries the team for corrected fee data. If the percentage check is passed, the program calculates achieved values, counts the run number for computation costs and displays the Achieved Values Table below.

```
                              ACHIEVED VALUES
 -----------------------------------------------------------------------------

 DP-3 *** DEVELOPMENT CONTRACT SUMMARY *** TEAM  1

                    INCENTIVE PROVISIONS          INCENTIVE ACHIEVEMENTS

  INCENTIVE       WORST    BEST     MAX FEE      ACHIEVED     FEE      FEE %
    AREA          VALUE    VALUE    ALLOWED        VALUE     EARNED    EARNED

 DEV. COST       $ 53.0M  $ 47.0M    5.0%       $ 50.64M   $ 0.984M   1.97%
 FLT TST COMPL    238 WK   202 WK    3.0%         199 WK   $ 1.500M   3.00%
 RELIABILITY       85.0%    91.0%    3.5%         83.74%   $ 0.0   M  0.0 %
 ACCURACY         160YDS   140YDS    3.5%         143YDS   $ 1.470M   2.94%

 TOTALS                             15.0%                 $  3.955M   7.91%

 TOTAL CONTRACT PRICE = $ 54.6M
```

```
          |IH0001A PAUSE;PRESS <<ENTER>> TO CONTINUE.
====><<ENTER>>
```

The "PAUSE" causes the screen to remain until the operator indicates he is ready to proceed.

<<ENTER>> symbolizes pressing the enter key which causes the program to start processing again.



```
                              MAIN MENU
----------------------------------------------------------------------

DO YOU WISH TO:
1. RECEIVE THE TABLE OF ACHIEVED VALUES.
                   *****WARNING*****AFTER THE TENTH DP- 3 RUN
                   AN ADDITIONAL DEVELOPMENT COST OF $100,000
                   WILL BE INCURRED FOR EACH ADDITIONAL RUN.
                   THIS IS RUN 2 OF DP- 3.
2. INPUT SELECTION MENU
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT
```

Sometimes the display will show "MORE" in the lower right margin of the screen. After several seconds the screen will scroll itself. PAUSE prevents this screen scrolling action. When "MORE" is displayed and the operator wishes to prevent the display from automatically scrolling, the display may be held by pressing <<ENTER>>. "HOLDING" will be displayed. To proceed from a "HOLDING" or "MORE" display, press the <<ALT>> and <<CLEAR>> keys simultaneously.

NOTE the count of calculation runs has changed. The program counts the occurrences of achieved value calculations. After ten calculations, the program adds a charge of $100,000 for each additional run.

```
====>2
```

The result of selecting 2 from the Main Menu, will be to reprint the Selection Menu from which a data item may be selected for change.

```
                              SELECTION MENU
 ------------------------------------------------------------------------
 NO.*ITEM******CURRENT VALUE @ DP-3    NO.*ITEM************CURRENT VALUE
  1. CONTRACTOR              :  1       16. GUIDANCE QUAL TESTS    :  6.00
  2. GUIDANCE APPROACH       :  1       17. FLIGHT TESTS           : 17.00
  3. GUIDANCE CONFIGURATION:   2        18. MAXIMUM COST        M$: 53.00
  4. MAINTAINABILITY ENG M$:   1.75     19. MINIMUM COST        M$: 47.00
  5. VALUE ENG            M$:  0.75     20. MAX COST INCENTIVE   %:  5.00
  6. PARALLEL DEVELOP     M$:  0.0      21. LATEST WEEK            :238.00
  7. MOTOR RELIABILITY     %: 92.00     22. EARLIEST WEEK          :202.00
  8. AIRFRAME RELIABILITY %: 97.00      23. MAX DELIVER INCENTIVE%:  3.00
  9. LAUNCHER RELIABILITY %: 98.50      24. MAXIMUM RELIABILITY  %: 91.00
 10. FIRE CONTROL ERROR YDS: 80.00      25. MINIMUM RELIABILITY  %: 85.00
 11. GUIDANCE ERROR     YDS: 70.00      26. MAX RELIABIL INCENTIV%:  3.50
 12. MOTOR QUAL TESTS      : 25.00      27. MAXIMUM ERROR      YDS:160.00
 13. AIRFRAME QUAL TESTS   :  6.00      28. MINIMUM ERROR      YDS:140.00
 14. LAUNCHER QUAL TESTS   :  3.00      29. MAX ERROR INCENTIVE  %:  3.50
 15. FIRE CONTROL QUAL TSTS:  3.00      30. WEEK FOR LOT 10        :337.00
             *****0. NONE*********31. CHANGE BY PAGE*****
            SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3.
```

```
====>6
         PREVIOUS VALUE 0.0; INPUT A VALUE?
====>2

         PARALLEL DEV FUNDS OF $ 2.00M
         ARE NOT REQUIRED AND HAVE BEEN RESET TO $0.0
```

The procedure for changing any data in the database is accomplished as follows.

To change the parallel development funds, choose item number 6 from the Selection Menu.

Parallel development funds are selected for change.

The new value input for parallel development funds is $2 million.

The value of parallel development funds at $2 million is not an acceptable option.

The program now cues the operator to the prob-

```
        DO YOU WISH TO CHANGE GUIDANCE CONFIGURATION:
        "Y"=YES OR "N"=NO?
====>N
```

lem by asking for a correction in the guidance configuration which prevented the desired change in parallel development funds.

After completing the change queries, the program reprints the Selection Menu with the changed values, ready for another selection.

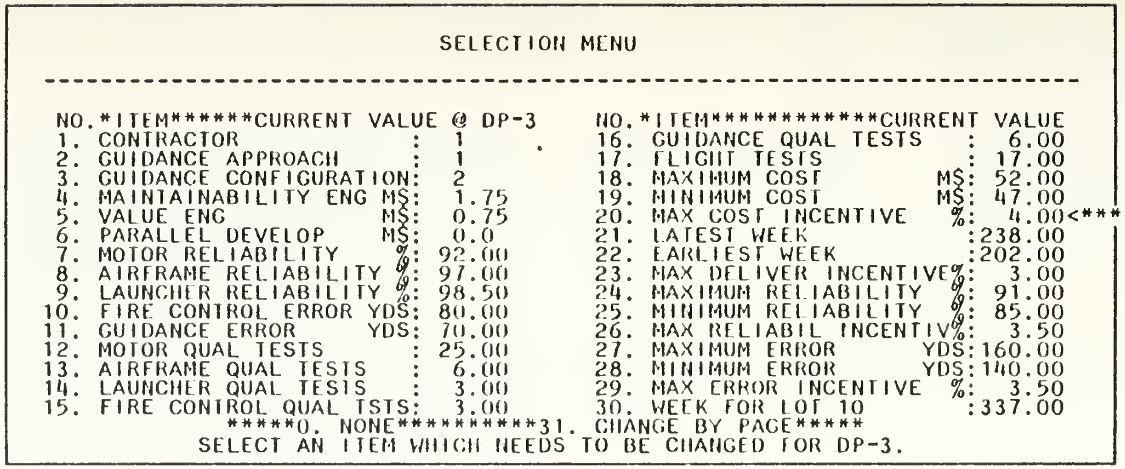
```
                                  SELECTION MENU

   ---------------------------------------------------------------------------

   NO.*ITEM******CURRENT VALUE @ DP-3    NO.*ITEM************CURRENT VALUE
    1. CONTRACTOR              :  1       16. GUIDANCE QUAL TESTS    :  6.00
    2. GUIDANCE APPROACH       :  1       17. FLIGHT TESTS           : 17.00
    3. GUIDANCE CONFIGURATION:  2         18. MAXIMUM COST         M$: 53.00
    4. MAINTAINABILITY ENG M$:  1.75      19. MINIMUM COST         M$: 47.00
    5. VALUE ENG           M$:  0.75      20. MAX COST INCENTIVE    %:  5.00<***** Notice the value.
    6. PARALLEL DEVELOP    M$:  0.0       21. LATEST WEEK            :238.00
    7. MOTOR RELIABILITY    %: 92.00      22. EARLIEST WEEK          :202.00
    8. AIRFRAME RELIABILITY %: 97.00      23. MAX DELIVER INCENTIVE%:  3.00
    9. LAUNCHER RELIABILITY %: 98.50      24. MAXIMUM RELIABILITY   %: 91.00
   10. FIRE CONTROL ERROR YDS: 80.00      25. MINIMUM RELIABILITY   %: 85.00
   11. GUIDANCE ERROR     YDS: 70.00      26. MAX RELIABIL INCENTIV%:  3.50
   12. MOTOR QUAL TESTS      : 25.00      27. MAXIMUM ERROR      YDS:160.00
   13. AIRFRAME QUAL TESTS   :  6.00      28. MINIMUM ERROR      YDS:140.00
   14. LAUNCHER QUAL TESTS   :  3.00      29. MAX ERROR INCENTIVE %:  3.50
   15. FIRE CONTROL QUAL TSTS:  3.00      30. WEEK FOR LOT 10        :337.00
                  *****0. NONE**********31. CHANGE BY PAGE*****
                SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3.
====>20

        PREVIOUS VALUE   5.00; INPUT A VALUE ?
====>4
```

Choose the item to be changed, the maximum fee for development cost incentive.

The previous value was 5%.

The desired new value is 4%.

The value is changed in the Selection Menu, and the menu is reprinted.



```
                         SELECTION MENU
------------------------------------------------------------------------------
 NO.*ITEM******CURRENT VALUE @ DP-3     NO.*ITEM***********CURRENT VALUE
  1. CONTRACTOR             :  1     .   16. GUIDANCE QUAL TESTS    :  6.00
  2. GUIDANCE APPROACH      :  1         17. FLIGHT TESTS           : 17.00
  3. GUIDANCE CONFIGURATION:  2          18. MAXIMUM COST         M$: 52.00
  4. MAINTAINABILITY ENG M$:  1.75       19. MINIMUM COST         M$: 47.00
  5. VALUE ENG           M$:  0.75       20. MAX COST INCENTIVE    %:  4.00<***** Notice the new value.
  6. PARALLEL DEVELOP    M$:  0.0        21. LATEST WEEK            :238.00
  7. MOTOR RELIABILITY    %: 92.00       22. EARLIEST WEEK          :202.00
  8. AIRFRAME RELIABILITY %: 97.00       23. MAX DELIVER INCENTIVE%:  3.00
  9. LAUNCHER RELIABILITY %: 98.50       24. MAXIMUM RELIABILITY   %: 91.00
 10. FIRE CONTROL ERROR YDS: 80.00       25. MINIMUM RELIABILITY   %: 85.00
 11. GUIDANCE ERROR     YDS: 70.00       26. MAX RELIABIL INCENTIV%:  3.50
 12. MOTOR QUAL TESTS      : 25.00       27. MAXIMUM ERROR       YDS:160.00
 13. AIRFRAME QUAL TESTS   :  6.00       28. MINIMUM ERROR       YDS:140.00
 14. LAUNCHER QUAL TESTS   :  3.00       29. MAX ERROR INCENTIVE   %:  3.50
 15. FIRE CONTROL QUAL TSTS:  3.00       30. WEEK FOR LOT 10        :337.00
              *****0. NONE*********31. CHANGE BY PAGE*****
          SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3.
```

```
====>0

      ***WARNING***TOTAL OF FEE PERCENTAGES IS LESS THAN 15%.
      THE VALUES YOU PREVIOUSLY SUBMITTED ARE:
      COST FEE INCENTIVE= 4.00
      DELIVERY DATE FEE INCENTIVE= 3.00
      RELIABILITY INCENTIVE FEE= 3.50
      ACCURACY INCENTIVE FEE= 3.50
      REDO YOUR INPUT.

      ENTER THE COST INCENTIVE FEE.
      PREVIOUS VALUE   4.00; INPUT A VALUE?
====>4
      YOU HAVE 11.00% OF THE INCENTIVE FEES REMAINING.
      ENTER THE DELIVERY DATE INCENTIVE FEE.
      PREVIOUS VALUE   3.00; INPUT A VALUE?
====>3.5
      YOU HAVE  7.50% OF THE INCENTIVE FEES REMAINING.
      ENTER THE RELIABILITY INCENTIVE FEE.
      PREVIOUS VALUE   3.50; INPUT A VALUE?
====>3.5
      YOU HAVE  4.00% OF THE INCENTIVE FEES REMAINING.
      ENTER THE ACCURACY INCENTIVE FEE.
      PREVIOUS VALUE   3.50; INPUT A VALUE?

====>3.5

      DO YOU WISH TO CONTINUE;"Y"=YES OR "N"=NO?
```

The result of selecting 0 from the Selection Menu will be to reprint the Main Menu.

Zero from the Selection Menu sends the program to the Main Menu.

Exit from the Selection Menu routine by entering '0'. The program will first evaluate the fee total percentage.

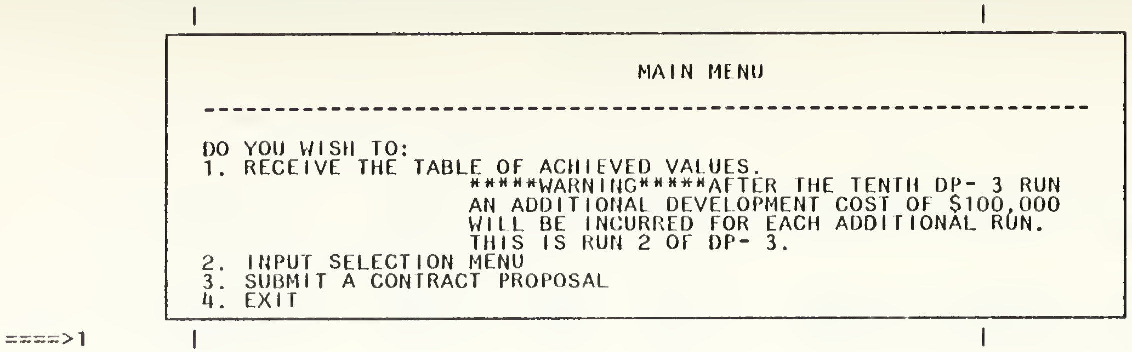
Whenever the fee total does not equal 15%, the program queries for correction.

These are the old values.

The program keeps track of the fee remainder. It tells how much fee remains to be allocated.

If the ACCURACY FEE entered is not equal to this last remainder, the program assumes a data entry error. It re-loops through the EXIT routine and can return to the percentage routine.

```
====>y      |                                                          |
            |***WARNING***TOTAL OF FEE PERCENTAGES IS LESS THAN 15%.   |Unsuccessful correction of the percentage
            |THE VALUES YOU PREVIOUSLY SUBMITTED ARE:                  |          fees returns the program to
            |COST FEE INCENTIVE= 4.00                                  |the percentage correction routine.
            |DELIVERY DATE FEE INCENTIVE= 3.00                         |
            |RELIABILITY INCENTIVE FEE= 3.50                           |These are the old values.
            |ACCURACY INCENTIVE FEE= 3.50                              |
            |REDO YOUR INPUT.                                          |
            |                                                          |
            |ENTER THE COST INCENTIVE FEE.                             |
            |PREVIOUS VALUE    4.00; INPUT A VALUE?                    |
====>4      |                                                          |The program keeps track of the fee remainder.
            |YOU HAVE 11.00% OF THE INCENTIVE FEES REMAINING.          |It tells how much fee remains to
            |ENTER THE DELIVERY DATE INCENTIVE FEE.                    |          be allocated.
            |PREVIOUS VALUE    3.00; INPUT A VALUE?                    |
====>3.5    |                                                          |
            |YOU HAVE  7.50% OF THE INCENTIVE FEES REMAINING.          |
            |ENTER THE RELIABILITY INCENTIVE FEE.                      |
            |PREVIOUS VALUE    3.50; INPUT A VALUE?                    |
====>3.5    |                                                          |
            |YOU HAVE  4.00% OF THE INCENTIVE FEES REMAINING.          |
            |ENTER THE ACCURACY INCENTIVE FEE.                         |Successful completion of the corrections
            |PREVIOUS VALUE    3.50; INPUT A VALUE?                    |          returns the program to the Main
            |                                                          |          Menu.
====>4.0    |                                                          |
```



```
                                        MAIN MENU
   ----------------------------------------------------------------------------

DO YOU WISH TO:
1. RECEIVE THE TABLE OF ACHIEVED VALUES.
                       *****WARNING*****AFTER THE TENTH DP- 3 RUN
                       AN ADDITIONAL DEVELOPMENT COST OF $100,000
                       WILL BE INCURRED FOR EACH ADDITIONAL RUN.
                       THIS IS RUN 2 OF DP- 3.
2. INPUT SELECTION MENU
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT
```

```
====>1
```

```
                              ACHIEVED VALUES
-------------------------------------------------------------------------------

DP-3 *** DEVELOPMENT CONTRACT SUMMARY *** TEAM  1

                   INCENTIVE PROVISIONS          INCENTIVE ACHIEVEMENTS

 INCENTIVE       WORST     BEST     MAX FEE     ACHIEVED     FEE       FEE %
   AREA          VALUE     VALUE    ALLOWED      VALUE      EARNED     EARNED

DEV. COST       $ 53.0M   $ 47.0M     4.0%     $ 50.81M   $ 0.730M    1.46%
FLT TST COMPL    238 WK    202 WK     3.5%       199 WK   $ 1.750M    3.50%
RELIABILITY       85.0%     91.0%     3.5%       83.81%   $ 0.0  M    0.0 %
ACCURACY         160YDS    140YDS     4.0%       143YDS   $ 1.680M    3.36%

TOTALS                               15.0%               $  4.161M    8.32%

TOTAL CONTRACT PRICE = $ 55.0M
```

```
                 |IH0001A PAUSE ;PRESS <<ENTER>> TO CONTINUE.     |The "PAUSE" causes the screen to retain the
====><<ENTER>>                                                    |        display until the operator indicates
                 |                                                |        he is ready to continue.
```

```
                              MAIN MENU
-----------------------------------------------------------------

DO YOU WISH TO:

1. RECEIVE THE TABLE OF ACHIEVED VALUES.
                    *****WARNING*****AFTER THE TENTH DP- 3 RUN
                    AN ADDITIONAL DEVELOPMENT COST OF $100,000
                    WILL BE INCURRED FOR EACH ADDITIONAL RUN.
                    THIS IS RUN 3 OF DP- 3.
2. INPUT SELECTION MENU
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT
```

```
====>3
```

From the Main menu, selection of 3 provides a printout of the database as the Report Submission Table.

```
                         PROPOSAL SUBMISSION TABLE
-------------------------------------------------------------------------

NO.*ITEM******CURRENT VALUE @ DP-3     NO.*ITEM***********CURRENT VALUE
 1. CONTRACTOR             :  1        16. GUIDANCE QUAL TESTS    :  6.00
 2. GUIDANCE APPROACH      :  1        17. FLIGHT TESTS           : 17.00
 3. GUIDANCE CONFIGURATION:  2         18. MAXIMUM COST        M$: 53.00
 4. MAINTAINABILITY ENG M$:  1.75      19. MINIMUM COST        M$: 47.00
 5. VALUE ENG           M$:  0.75      20. MAX COST INCENTIVE   %:  4.00
 6. PARALLEL DEVELOP    M$:  0.0       21. LATEST WEEK            :238.00
 7. MOTOR RELIABILITY    %: 92.00      22. EARLIEST WEEK          :202.00
 8. AIRFRAME RELIABILITY %: 97.00      23. MAX DELIVER INCENTIVE%:  3.50
 9. LAUNCHER RELIABILITY %: 98.50      24. MAXIMUM RELIABILITY  %: 91.00
10. FIRE CONTROL ERROR YDS: 80.00      25. MINIMUM RELIABILITY  %: 85.00
11. GUIDANCE ERROR     YDS: 70.00      26. MAX RELIABIL INCENTIV%:  3.50
12. MOTOR QUAL TESTS      : 25.00      27. MAXIMUM ERROR      YDS:160.00
13. AIRFRAME QUAL TESTS   :  6.00      28. MINIMUM ERROR      YDS:140.00
14. LAUNCHER QUAL TESTS   :  3.00      29. MAX ERROR INCENTIVE  %:  4.00
15. FIRE CONTROL QUAL TSTS:  3.00      30. WEEK FOR LOT 10        :337.00
****DO YOU WISH TO ENTER THIS DATA AS YOUR PROPOSED DP- 3 DECISION?****
       *****CAUTION*****CAUTION*****CAUTION*****
                        YOU MAY ONLY INPUT A DECISION ONCE.
ENTER YOUR TEAM SECURITY CODE IF YOU WISH TO COMMIT TO A DECISION;
OR "CONT".
```

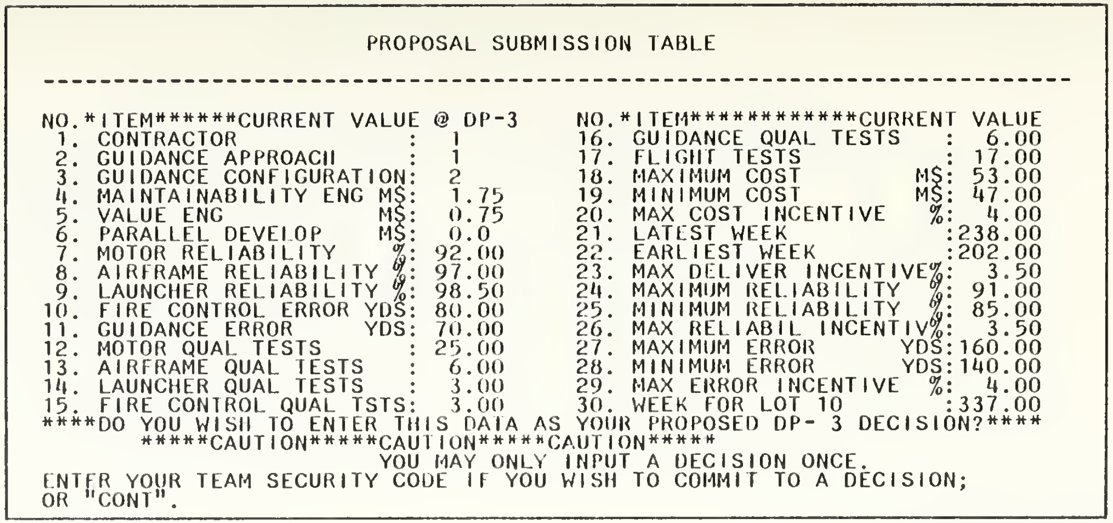
```
====>cont
```

The result of entering "continue' is to return and reprint the Main Menu.

Continue returns to the Main Menu without submitting a report.

```
                        MAIN MENU
 ---------------------------------------------------------

 DO YOU WISH TO:

 1. RECEIVE THE TABLE OF ACHIEVED VALUES.
                   *****WARNING*****AFTER THE TENTH DP- 3 RUN
                   AN ADDITIONAL DEVELOPMENT COST OF $100,000
                   WILL BE INCURRED FOR EACH ADDITIONAL RUN.
                   THIS IS RUN 3 OF DP- 3.
 2. INPUT SELECTION MENU
 3. SUBMIT A CONTRACT PROPOSAL
 4. EXIT

====>3
```

Provides a printout of the database to verify the proposed values being submitted.



```
                          PROPOSAL SUBMISSION TABLE
---------------------------------------------------------------------------
NO.*ITEM******CURRENT VALUE @ DP-3     NO.*ITEM***********CURRENT VALUE
 1. CONTRACTOR            :  1         16. GUIDANCE QUAL TESTS    :  6.00
 2. GUIDANCE APPROACH     :  1         17. FLIGHT TESTS           : 17.00
 3. GUIDANCE CONFIGURATION:  2         18. MAXIMUM COST         M$: 53.00
 4. MAINTAINABILITY ENG M$:   1.75     19. MINIMUM COST         M$: 47.00
 5. VALUE ENG           M$:   0.75     20. MAX COST INCENTIVE    %:  4.00
 6. PARALLEL DEVELOP    M$:   0.0      21. LATEST WEEK            :238.00
 7. MOTOR RELIABILITY    %: 92.00      22. EARLIEST WEEK          :202.00
 8. AIRFRAME RELIABILITY %: 97.00      23. MAX DELIVER INCENTIVE%:  3.50
 9. LAUNCHER RELIABILITY %: 98.50      24. MAXIMUM RELIABILITY   %: 91.00
10. FIRE CONTROL ERROR YDS: 80.00      25. MINIMUM RELIABILITY   %: 85.00
11. GUIDANCE ERROR     YDS: 70.00      26. MAX RELIABIL INCENTIV%:  3.50
12. MOTOR QUAL TESTS      : 25.00      27. MAXIMUM ERROR       YDS:160.00
13. AIRFRAME QUAL TESTS   :  6.00      28. MINIMUM ERROR       YDS:140.00
14. LAUNCHER QUAL TESTS   :  3.00      29. MAX ERROR INCENTIVE   %:  4.00
15. FIRE CONTROL QUAL TSTS:  3.00      30. WEEK FOR LOT 10        :337.00
****DO YOU WISH TO ENTER THIS DATA AS YOUR PROPOSED DP- 3 DECISION?****
       *****CAUTION*****CAUTION*****CAUTION*****
                        YOU MAY ONLY INPUT A DECISION ONCE.
ENTER YOUR TEAM SECURITY CODE IF YOU WISH TO COMMIT TO A DECISION;
OR "CONT".
====>01kline |                                                        |
```

```
                       REPORT ACCEPTED VALUES TABLE

 ------------------------------------------------------------------------

         ******THE FOLLOWING PARAMETERS WERE ACCEPTED******
         AS TEAM  1 DP- 3 PROPOSED INPUT TO THE CONTRACTOR.
NO.*ITEM******CURRENT VALUE @ DP-3     NO.*ITEM*************CURRENT VALUE
 1. CONTRACTOR            :  1          16. GUIDANCE QUAL TESTS   :  6.00
 2. GUIDANCE APPROACH     :  1          17. FLIGHT TESTS          : 17.00
 3. GUIDANCE CONFIGURATION:  2          18. MAXIMUM COST        M$: 53.00
 4. MAINTAINABILITY ENG M$:  1.75       19. MINIMUM COST        M$: 47.00
 5. VALUE ENG           M$:  0.75       20. MAX COST INCENTIVE   %:  4.00
 6. PARALLEL DEVELOP    M$:  0.0        21. LATEST WEEK           :238.00
 7. MOTOR RELIABILITY    %: 92.00       22. EARLIEST WEEK         :202.00
 8. AIRFRAME RELIABILITY %: 97.00       23. MAX DELIVER INCENTIVE%:  3.50
 9. LAUNCHER RELIABILITY %: 98.50       24. MAXIMUM RELIABILITY  %: 91.00
10. FIRE CONTROL ERROR YDS: 80.00       25. MINIMUM RELIABILITY  %: 85.00
11. GUIDANCE ERROR     YDS: 70.00       26. MAX RELIABIL INCENTIV%:  3.50
12. MOTOR QUAL TESTS      : 25.00       27. MAXIMUM ERROR      YDS:160.00
13. AIRFRAME QUAL TESTS   :  6.00       28. MINIMUM ERROR      YDS:140.00
14. LAUNCHER QUAL TESTS   :  3.00       29. MAX ERROR INCENTIVE  %:  4.00
15. FIRE CONTROL QUAL TSTS:  3.00       30. WEEK FOR LOT 10       :337.00
         *****SEE YOUR MONITOR IF YOU HAVE A PROBLEM.******
```

Entering the team's security code will rewrite the file DATABASE TEAMxx.

From the Main menu, selection of option 3 directs the program to the report submission routine. This routine determines the DP report last submitted. It then displays the current database selections in a Report Submission Table as the offering for the next report. Team approval of the Submission Table is made by entering the team's security code. The program can sequence the team through

the stages of contract submission including DP-3 and 4's proposed and final contract reports.

```
====><<ENTER>>     IH0001A PAUSE ;PRESS <<ENTER>> TO CONTINUE.
```

Pressing <<ENTER>> returns the program to the Main Menu display.

```
                                      MAIN MENU
     ----------------------------------------------------------------

     DO YOU WISH TO:

     1. RECEIVE THE TABLE OF ACHIEVED VALUES.
                           *****WARNING*****AFTER THE TENTH DP- 3 RUN
                           AN ADDITIONAL DEVELOPMENT COST OF $100,000
                           WILL BE INCURRED FOR EACH ADDITIONAL RUN.
                           THIS IS RUN 3 OF DP- 3.
     2. INPUT SELECTION MENU
     3. SUBMIT A CONTRACT PROPOSAL
     4. EXIT
```

```
====>4
                   PUN FILE 8092  TO  0543P    COPY 001    NOHOLD
                   PUN FILE 8094  TO  0543P    COPY 001    NOHOLD
                   ANOTHER RUN, 'Y'/'N'?
====>Y
```

Exits from the program.
The data files are now transmitted to the monitor.
The LINKPROJ EXEC program allows the user to reloop to the program start and begin again.

```
                   BEGIN RECORDING OF TERMINAL SESSION
                   * NOTE!: YOU ARE NOW LINKED TO PROJMNG ON YOUR 192 DISK
                   MODE C
                   * NOTE!: PRESS PF09 TO BREAK THE LINK
                   FILE 'DATAINST DISKNUM A' NOT FOUND
                   FILE 'DATAFILE TEAM01 C' NOT FOUND.
                   FILE 'DATACODE TEAM01 C' NOT FOUND.


                   WAIT FOR "EXECUTION TO BEGIN"
                   EXECUTION BEGINS...
```

another session.
This time DATAFILE TEAM01 A is found and so is DATACODE TEAM01 A. They now exist on your A disk.

```
*************************************************************
         TO TERMINATE THE PROGRAM AT ANY POINT,
     TYPE "E"=EXIT IN RESPONSE TO ANY "YES/NO" QUERY.
*************************************************************
***CAUTION***IN RESPONSE TO A QUERY, PRESSING <<ENTER>>
     WITHOUT PROVIDING DATA WILL DUMP THE PROGRAM.
*************************************************************

PLEASE ENTER YOUR TEAM SECURITY CODE.
====>00NPS001
```

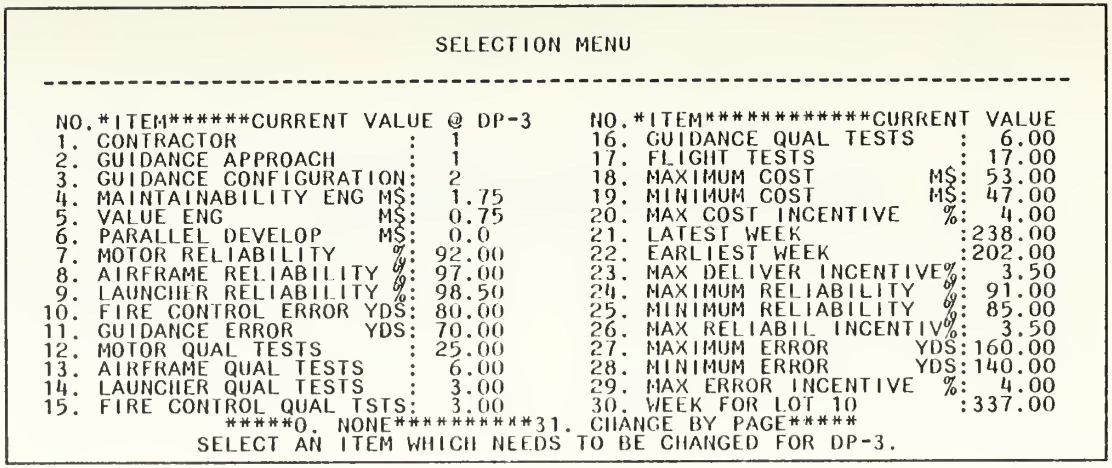
If you misenter your previously selected security code the following message will appear.

```
YOU HAVE ATTEMPTED TO ENTER A TEAM FILE
WITHOUT THE PROPER SECURITY CODE.
DO YOU WISH TO CONTINUE: "Y"=YES OR "N"=NO?
====>y
```

In case of an error you may enter 'y' to return to the TEAM query.)

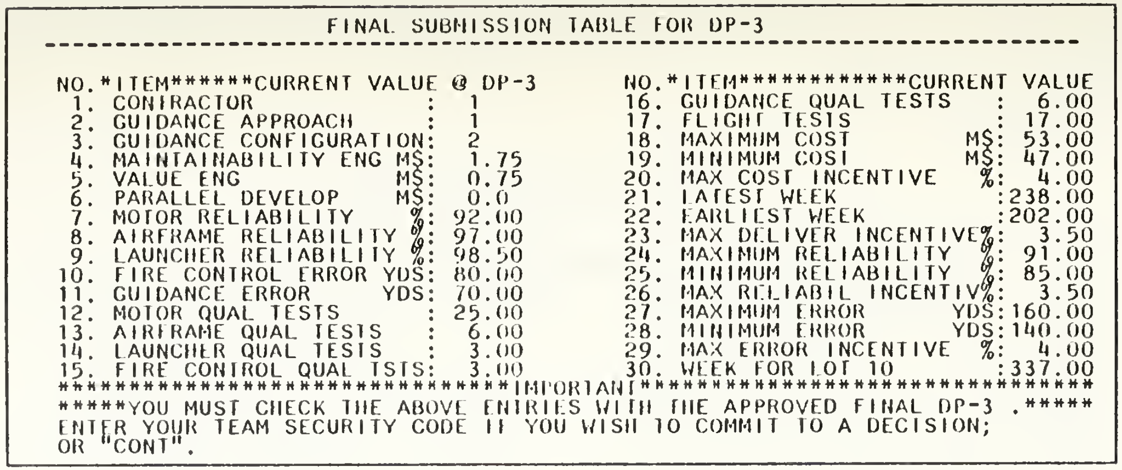
```
PLEASE ENTER YOUR TEAM SECURITY CODE.
====>01kline
```

The program will now proceed to check the database for previous information. It will display the data in a SELECTION MENU.



```
                              SELECTION MENU
--------------------------------------------------------------------------------
 NO.*ITEM******CURRENT VALUE @ DP-3    NO.*ITEM************CURRENT VALUE
  1. CONTRACTOR            :  1        16. GUIDANCE QUAL TESTS    :  6.00
  2. GUIDANCE APPROACH     :  1        17. FLIGHT TESTS           : 17.00
  3. GUIDANCE CONFIGURATION:  2        18. MAXIMUM COST         M$: 53.00
  4. MAINTAINABILITY ENG M$:  1.75     19. MINIMUM COST         M$: 47.00
  5. VALUE ENG           M$:  0.75     20. MAX COST INCENTIVE    %:  4.00
  6. PARALLEL DEVELOP    M$:  0.0      21. LATEST WEEK            :238.00
  7. MOTOR RELIABILITY    %: 92.00     22. EARLIEST WEEK          :202.00
  8. AIRFRAME RELIABILITY %: 97.00     23. MAX DELIVER INCENTIVE%:  3.50
  9. LAUNCHER RELIABILITY %: 98.50     24. MAXIMUM RELIABILITY   %: 91.00
 10. FIRE CONTROL ERROR YDS: 80.00     25. MINIMUM RELIABILITY   %: 85.00
 11. GUIDANCE ERROR     YDS: 70.00     26. MAX RELIABIL INCENTIV%:  3.50
 12. MOTOR QUAL TESTS      : 25.00     27. MAXIMUM ERROR      YDS:160.00
 13. AIRFRAME QUAL TESTS   :  6.00     28. MINIMUM ERROR      YDS:140.00
 14. LAUNCHER QUAL TESTS   :  3.00     29. MAX ERROR INCENTIVE   %:  4.00
 15. FIRE CONTROL QUAL TSTS:  3.00     30. WEEK FOR LOT 10        :337.00
                 *****0. NONE*********31. CHANGE BY PAGE*****
              SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-3.
====>0
```

```
                                MAIN MENU
 ---------------------------------------------------------------

 DO YOU WISH TO:

 1. RECEIVE THE TABLE OF ACHIEVED VALUES.
                    *****WARNING*****AFTER THE TENTH DP- 3 RUN
                    AN ADDITIONAL DEVELOPMENT COST OF $100,000
                    WILL BE INCURRED FOR EACH ADDITIONAL RUN.
                    THIS IS RUN 3 OF DP- 3.
 2. INPUT SELECTION MENU
 3. SUBMIT A CONTRACT PROPOSAL
 4. EXIT
====>3
```



```
                         FINAL SUBMISSION TABLE FOR DP-3
-----------------------------------------------------------------------------

NO.*ITEM******CURRENT VALUE @ DP-3      NO.*ITEM***********CURRENT VALUE
 1. CONTRACTOR              :  1        16. GUIDANCE QUAL TESTS    :  6.00
 2. GUIDANCE APPROACH       :  1        17. FLIGHT TESTS           : 17.00
 3. GUIDANCE CONFIGURATION:    2        18. MAXIMUM COST         M$: 53.00
 4. MAINTAINABILITY ENG M$:    1.75     19. MINIMUM COST         M$: 47.00
 5. VALUE ENG           M$:    0.75     20. MAX COST INCENTIVE    %:  4.00
 6. PARALLEL DEVELOP    M$:    0.0      21. LATEST WEEK            :238.00
 7. MOTOR RELIABILITY    %: 92.00       22. EARLIEST WEEK          :202.00
 8. AIRFRAME RELIABILITY %: 97.00       23. MAX DELIVER INCENTIVE%:  3.50
 9. LAUNCHER RELIABILITY %: 98.50       24. MAXIMUM RELIABILITY   %: 91.00
10. FIRE CONTROL ERROR YDS: 80.00       25. MINIMUM RELIABILITY   %: 85.00
11. GUIDANCE ERROR     YDS: 70.00       26. MAX RELIABIL INCENTIV%:  3.50
12. MOTOR QUAL TESTS      : 25.00       27. MAXIMUM ERROR       YDS:160.00
13. AIRFRAME QUAL TESTS   :  6.00       28. MINIMUM ERROR       YDS:140.00
14. LAUNCHER QUAL TESTS   :  3.00       29. MAX ERROR INCENTIVE  %:  4.00
15. FIRE CONTROL QUAL TSTS:  3.00       30. WEEK FOR LOT 10        :337.00
*******************************IMPORTANT*******************************
*****YOU MUST CHECK THE ABOVE ENTRIES WITH THE APPROVED FINAL DP-3 .*****
ENTER YOUR TEAM SECURITY CODE IF YOU WISH TO COMMIT TO A DECISION;
OR "CONT".
```

The following sequence will demonstrate

- first the filing of a final DP-3 contract proposal;
- second we will exit from the
- program to reenter the program at DP-4
- fourth we will review the DP-4 Selection Menu as developed from the current DP-3 data base;
- next the changing of the paired sets of data for expected minimum and maximum reliability.
- sixth DP-4 will be evaluated with

```
====>01kline
```

- a table of achieved values;and finally, we will exit from the program.

Entering your team's security code will submit the final DP-3 report to the monitor.

```
                         REPORT ACCEPTED VALUES TABLE

 ----------------------------------------------------------------------

              ******THE FOLLOWING PARAMETERS WERE ACCEPTED******
              AS TEAM  1 DP- 3 FINAL*** INPUT TO THE CONTRACTOR.
 NO.*ITEM******CURRENT VALUE @ DP-3      NO.*ITEM***********CURRENT VALUE
  1. CONTRACTOR            :  1          16. GUIDANCE QUAL TESTS   :   6.00
  2. GUIDANCE APPROACH     :  1          17. FLIGHT TESTS          :  17.00
  3. GUIDANCE CONFIGURATION:  2          18. MAXIMUM COST        M$:  53.00
  4. MAINTAINABILITY ENG M$:   1.75      19. MINIMUM COST        M$:  47.00
  5. VALUE ENG           M$:   0.75      20. MAX COST INCENTIVE   %:   4.00
  6. PARALLEL DEVELOP    M$:   0.0       21. LATEST WEEK            :238.00
  7. MOTOR RELIABILITY    %:  92.00      22. EARLIEST WEEK          :202.00
  8. AIRFRAME RELIABILITY %:  97.00      23. MAX DELIVER INCENTIVE%:   3.50
  9. LAUNCHER RELIABILITY %:  98.50      24. MAXIMUM RELIABILITY  %:  91.00
 10. FIRE CONTROL ERROR YDS:  80.00      25. MINIMUM RELIABILITY  %:  85.00
 11. GUIDANCE ERROR     YDS:  70.00      26. MAX RELIABIL INCENTIV%:   3.50
 12. MOTOR QUAL TESTS      :  25.00      27. MAXIMUM ERROR      YDS:160.00
 13. AIRFRAME QUAL TESTS   :   6.00      28. MINIMUM ERROR      YDS:140.00
 14. LAUNCHER QUAL TESTS   :   3.00      29. MAX ERROR INCENTIVE  %:   4.00
 15. FIRE CONTROL QUAL TSTS:   3.00      30. WEEK FOR LOT 10        :337.00
              *****SEE YOUR MONITOR IF YOU HAVE A PROBLEM.******

IHO001A PAUSE ;PRESS <<ENTER>> TO CONTINUE.
====><<ENTER>>
```

```
                                MAIN MENU
--------------------------------------------------------------------

DO YOU WISH TO:
1. RECEIVE THE TABLE OF ACHIEVED VALUES.
                    *****WARNING*****AFTER THE TENTH DP-4 RUN
                    AN ADDITIONAL DEVELOPMENT COST OF $100,000
                    WILL BE INCURRED FOR EACH ADDITIONAL RUN.
                    THIS IS RUN  1 OF DP-4.                <*****

2. INPUT SELECTION MENU.
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT
```

NOTE the change in DP- level in this block.

```
====>4

PUN FILE 8092  TO  0543P    COPY 001    NOHOLD
PUN FILE 8094  TO  0543P    COPY 001    NOHOLD
ANOTHER RUN, 'Y'/'N'?
====>Y
```

This step will stop the program and return control to the EXEC, PROJMNG EXEC.

The data files are now transmitted to the monitor.

The LINKPROJ EXEC program allows the user to reloop to the program start and begin again.

DP-4

The following are DP-4 procedures.
Logon as you did for DP-3.

```
BEGIN RECORDING OF TERMINAL SESSION
* NOTE!: YOU ARE NOW LINKED TO PROJMNG ON YOUR 192 DISK
MODE C
* NOTE!: PRESS PF09 TO BREAK THE LINK
FILE 'DATAINST DISKNUM A' NOT FOUND
FILE 'DATAFILE TEAM01 C' NOT FOUND.
FILE 'DATACODE TEAM01 C' NOT FOUND.
PRESS <<ENTER> WHEN YOU ARE READY TO CONTINUE
```

another session.
This time DATAFILE TEAM01 A is found; and, so is DATACODE TEAM01 A. They now exist on your A disk.

```
WAIT FOR "EXECUTION TO BEGIN"
EXECUTION BEGINS...

*******************************************************
          TO TERMINATE THE PROGRAM AT ANY POINT,
      TYPE "E"=EXIT IN RESPONSE TO ANY "YES/NO" QUERY.
*******************************************************
***CAUTION***IN RESPONSE TO A QUERY, PRESSING <<ENTER>>
      WITHOUT PROVIDING DATA WILL DUMP THE PROGRAM.
*******************************************************

PLEASE ENTER YOUR TEAM SECURITY CODE.
====>01kline
```

The program will now proceed to check the database for previous information. It will display the data in a SELECTION MENU.

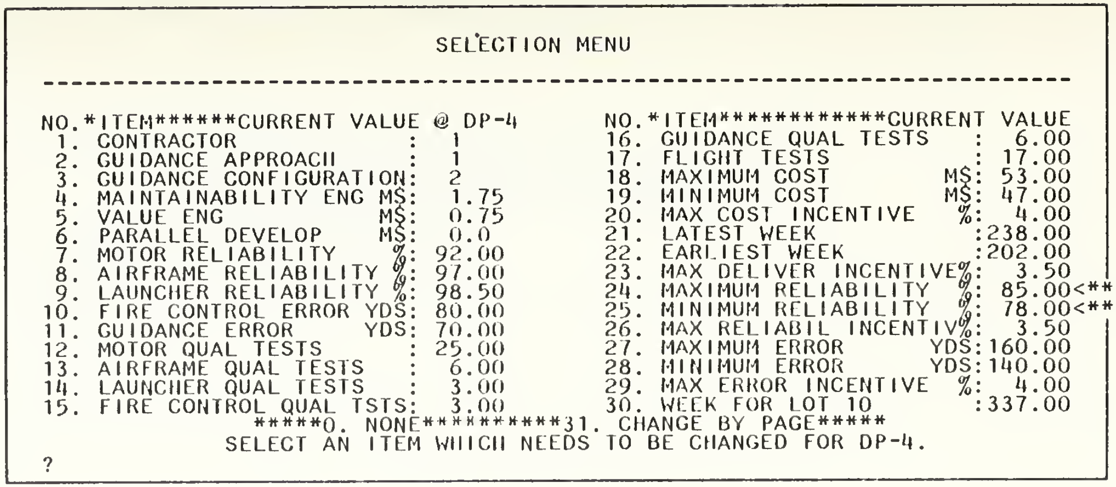
```
                              SELECTION MENU

 ---------------------------------------------------------------------------

 NO.*ITEM******CURRENT VALUE @ DP-4      NO.*ITEM***********CURRENT VALUE
  1. CONTRACTOR             :  1          16. GUIDANCE QUAL TESTS    :  6.00
  2. GUIDANCE APPROACH      :  1          17. FLIGHT TESTS           : 17.00
  3. GUIDANCE CONFIGURATION:  2           18. MAXIMUM COST         M$: 53.00
  4. MAINTAINABILITY ENG M$:  1.75        19. MINIMUM COST         M$: 47.00
  5. VALUE ENG           M$:  0.75        20. MAX COST INCENTIVE    %:  4.00
  6. PARALLEL DEVELOP    M$:  0.0         21. LATEST WEEK            :238.00
  7. MOTOR RELIABILITY    %: 92.00        22. EARLIEST WEEK          :202.00
  8. AIRFRAME RELIABILITY %: 97.00        23. MAX DELIVER INCENTIVE%:  3.50
  9. LAUNCHER RELIABILITY %: 98.50        24. MAXIMUM RELIABILITY   %: 91.00<****Notice the value.
 10. FIRE CONTROL ERROR YDS: 80.00        25. MINIMUM RELIABILITY   %: 85.00
 11. GUIDANCE ERROR     YDS: 70.00        26. MAX RELIABIL INCENTIV%:  3.50
 12. MOTOR QUAL TESTS      : 25.00        27. MAXIMUM ERROR       YDS:160.00
 13. AIRFRAME QUAL TESTS   :  6.00        28. MINIMUM ERROR       YDS:140.00
 14. LAUNCHER QUAL TESTS   :  3.00        29. MAX ERROR INCENTIVE   %:  4.00
 15. FIRE CONTROL QUAL TSTS:  3.00        30. WEEK FOR LOT 10        :337.00
                 *****0. NONE**********31. CHANGE BY PAGE*****
              SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-4.
```

```
====>24      PREVIOUS VALUE  91.00; INPUT A VALUE
             ?
====>85      DO YOU DESIRE TO CHANGE MINIMUM RELIABILITY, ALSO.

====>y       PREVIOUS VALUE  85.00; INPUT A VALUE
             ?
====>78
```

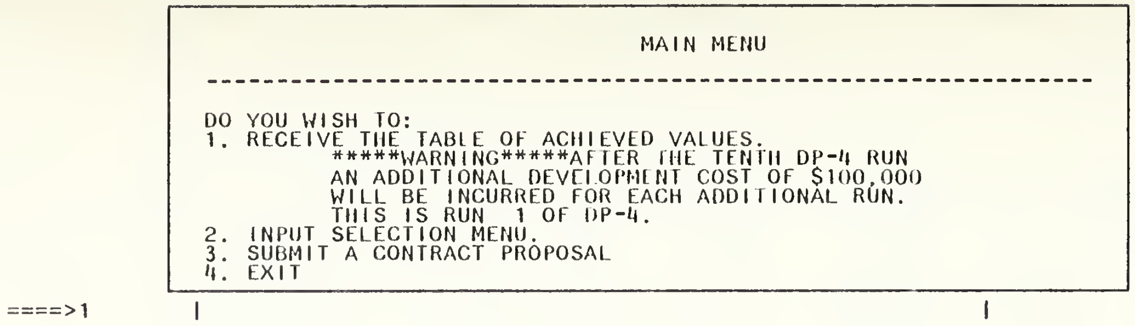
We are changing the maximum reliability.

The new value for maximum reliability is 85.

The program pairs the maximums and minimums for queries. If one is changed the program will ask if you wish to change the other.



```
                              SELECTION MENU

 ---------------------------------------------------------------------------

 NO.*ITEM******CURRENT VALUE @ DP-4      NO.*ITEM***********CURRENT VALUE
  1. CONTRACTOR             :  1         16. GUIDANCE QUAL TESTS   :  6.00
  2. GUIDANCE APPROACH      :  1         17. FLIGHT TESTS          : 17.00
  3. GUIDANCE CONFIGURATION:  2          18. MAXIMUM COST        M$: 53.00
  4. MAINTAINABILITY ENG M$:  1.75       19. MINIMUM COST        M$: 47.00
  5. VALUE ENG           M$:  0.75       20. MAX COST INCENTIVE   %:  4.00
  6. PARALLEL DEVELOP    M$:  0.0        21. LATEST WEEK           :238.00
  7. MOTOR RELIABILITY    %: 92.00       22. EARLIEST WEEK         :202.00
  8. AIRFRAME RELIABILITY %: 97.00       23. MAX DELIVER INCENTIVE%:  3.50
  9. LAUNCHER RELIABILITY %: 98.50       24. MAXIMUM RELIABILITY  %: 85.00<****Notice the new value.
 10. FIRE CONTROL ERROR YDS: 80.00       25. MINIMUM RELIABILITY  %: 78.00<****
 11. GUIDANCE ERROR     YDS: 70.00       26. MAX RELIABIL INCENTIV%:  3.50
 12. MOTOR QUAL TESTS      : 25.00       27. MAXIMUM ERROR      YDS:160.00
 13. AIRFRAME QUAL TESTS   :  6.00       28. MINIMUM ERROR      YDS:140.00
 14. LAUNCHER QUAL TESTS   :  3.00       29. MAX ERROR INCENTIVE  %:  4.00
 15. FIRE CONTROL QUAL TSTS:  3.00       30. WEEK FOR LOT 10       :337.00
                 *****0. NONE**********31. CHANGE BY PAGE*****
               SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-4.
 ?
====>0
```



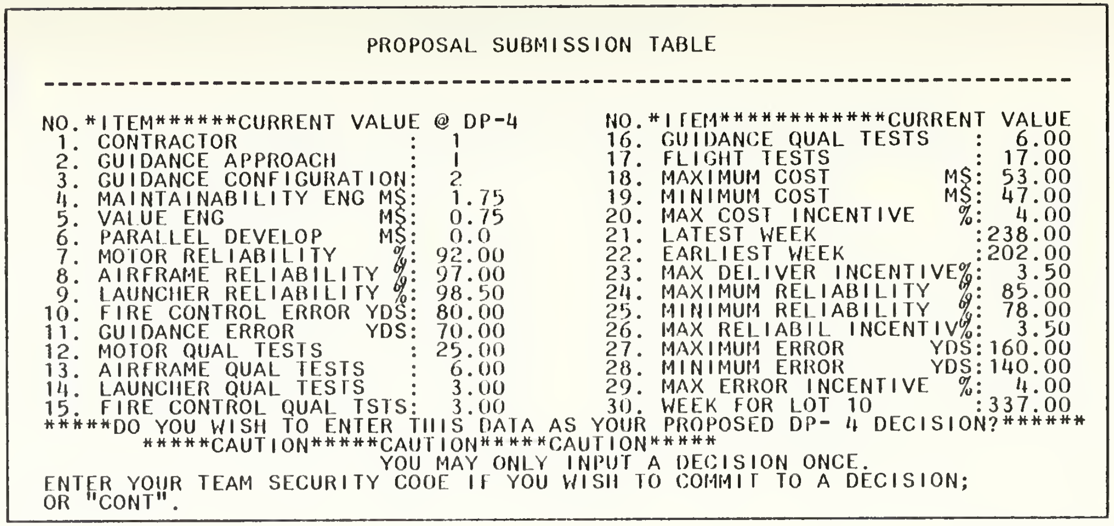
```
                                MAIN MENU
------------------------------------------------------------------------

DO YOU WISH TO:
1. RECEIVE THE TABLE OF ACHIEVED VALUES.
         *****WARNING*****AFTER THE TENTH DP-4 RUN
         AN ADDITIONAL DEVELOPMENT COST OF $100,000
         WILL BE INCURRED FOR EACH ADDITIONAL RUN.
         THIS IS RUN  1 OF DP-4.
2. INPUT SELECTION MENU.
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT
====>1       |                                                       |
```

```
                         ACHIEVED VALUES
--------------------------------------------------------------------------

DP-4 *** DEVELOPMENT CONTRACT SUMMARY *** TEAM  1

                   INCENTIVE PROVISIONS          INCENTIVE ACHIEVEMENTS

 INCENTIVE      WORST     BEST     MAX FEE     ACHIEVED     FEE      FEE %
   AREA         VALUE     VALUE    ALLOWED      VALUE      EARNED    EARNED

DEV. COST      $ 53.0M  $ 47.0M     4.0%     $ 50.87M   $ 0.710M    1.42%
FLT TST COMPL   238 WK   202 WK     3.5%       199 WK   $ 1.750M    3.50%
RELIABILITY      78.0%    85.0%     3.5%       83.81%   $ 1.452M    2.90%
ACCURACY        160YDS   140YDS     4.0%       143YDS   $ 1.681M    3.36%

TOTALS                             15.0%                $ 5.593M   11.19%

TOTAL CONTRACT PRICE = $ 56.5M
```

```
IH0001A PAUSE ;PRESS <<ENTER>> TO CONTINUE.
====><<ENTER>>
```

The "PAUSE" causes the screen to retain the display until the operator is ready to continue.

```
                              MAIN MENU
-----------------------------------------------------------------

DO YOU WISH TO:
1. RECEIVE THE TABLE OF ACHIEVED VALUES.
         *****WARNING*****AFTER THE TENTH DP-4 RUN
         AN ADDITIONAL DEVELOPMENT COST OF $100,000
         WILL BE INCURRED FOR EACH ADDITIONAL RUN.
         THIS IS RUN  2 OF DP-4.
2. INPUT SELECTION MENU.
3. SUBMIT A CONTRACT PROPOSAL
4. EXIT

====>3   |                                                      |
```



```
                          PROPOSAL SUBMISSION TABLE

----------------------------------------------------------------------------

NO.*ITEM******CURRENT VALUE @ DP-4      NO.*ITEM************CURRENT VALUE
 1. CONTRACTOR             :  1          16. GUIDANCE QUAL TESTS    :  6.00
 2. GUIDANCE APPROACH      :  1          17. FLIGHT TESTS           : 17.00
 3. GUIDANCE CONFIGURATION:  2           18. MAXIMUM COST         M$: 53.00
 4. MAINTAINABILITY ENG M$:  1.75        19. MINIMUM COST         M$: 47.00
 5. VALUE ENG           M$:  0.75        20. MAX COST INCENTIVE    %:  4.00
 6. PARALLEL DEVELOP    M$:  0.0         21. LATEST WEEK            :238.00
 7. MOTOR RELIABILITY    %: 92.00        22. EARLIEST WEEK          :202.00
 8. AIRFRAME RELIABILITY %: 97.00        23. MAX DELIVER INCENTIVE%:   3.50
 9. LAUNCHER RELIABILITY %: 98.50        24. MAXIMUM RELIABILITY   %: 85.00
10. FIRE CONTROL ERROR YDS: 80.00        25. MINIMUM RELIABILITY   %: 78.00
11. GUIDANCE ERROR     YDS: 70.00        26. MAX RELIABIL INCENTIV%:   3.50
12. MOTOR QUAL TESTS      : 25.00        27. MAXIMUM ERROR       YDS:160.00
13. AIRFRAME QUAL TESTS   :  6.00        28. MINIMUM ERROR       YDS:140.00
14. LAUNCHER QUAL TESTS   :  3.00        29. MAX ERROR INCENTIVE   %:  4.00
15. FIRE CONTROL QUAL TSTS:  3.00        30. WEEK FOR LOT 10        :337.00
*****DO YOU WISH TO ENTER THIS DATA AS YOUR PROPOSED DP- 4 DECISION?******
       *****CAUTION*****CAUTION*****CAUTION*****
                         YOU MAY ONLY INPUT A DECISION ONCE.
ENTER YOUR TEAM SECURITY CODE IF YOU WISH TO COMMIT TO A DECISION;
OR "CONT".
```

```
====>01kline
```

Entering the team's security code completes the filing of the new report to the monitor.

```
                              ACCEPTED FINAL DP-4

 -------------------------------------------------------------------------

              ******THE FOLLOWING PARAMETERS WERE ACCEPTED******
              AS TEAM  1 DP- 4 PROPOSED INPUT TO THE CONTRACTOR.
 NO.*ITEM******CURRENT VALUE @ DP-4       NO.*ITEM************CURRENT VALUE
  1. CONTRACTOR              :  1         16. GUIDANCE QUAL TESTS    :  6.00
  2. GUIDANCE APPROACH       :  1         17. FLIGHT TESTS           : 17.00
  3. GUIDANCE CONFIGURATION:   2          18. MAXIMUM COST         M$: 53.00
  4. MAINTAINABILITY ENG M$:   1.75       19. MINIMUM COST         M$: 47.00
  5. VALUE ENG            M$:  0.75       20. MAX COST INCENTIVE    %:  4.00
  6. PARALLEL DEVELOP     M$:  0.0        21. LATEST WEEK            :238.00
  7. MOTOR RELIABILITY     %: 92.00       22. EARLIEST WEEK          :202.00
  8. AIRFRAME RELIABILITY  %: 97.00       23. MAX DELIVER INCENTIVE%:  3.50
  9. LAUNCHER RELIABILITY  %: 98.50       24. MAXIMUM RELIABILITY   %: 85.00
 10. FIRE CONTROL ERROR YDS: 80.00        25. MINIMUM RELIABILITY   %: 78.00
 11. GUIDANCE ERROR     YDS: 70.00        26. MAX RELIABIL INCENTIV%:  3.50
 12. MOTOR QUAL TESTS      : 25.00        27. MAXIMUM ERROR      YDS:160.00
 13. AIRFRAME QUAL TESTS   :  6.00        28. MINIMUM ERROR      YDS:140.00
 14. LAUNCHER QUAL TESTS   :  3.00        29. MAX ERROR INCENTIVE   %:  4.00
 15. FIRE CONTROL QUAL TSTS:  3.00        30. WEEK FOR LOT 10        :337.00
              *****SEE YOUR MONITOR IF YOU HAVE A PROBLEM.******

      | IHO001A PAUSE ;PRESS <<ENTER>> TO CONTINUE.                     |
====><<ENTER>>
```

```
                             MAIN MENU
   ------------------------------------------------------------------

   DO YOU WISH TO:
   1. RECEIVE THE TABLE OF ACHIEVED VALUES.
            *****WARNINC*****AFTER THE TENTH DP-4 RUN
            AN ADDITIONAL DEVELOPMENT COST OF $100,000
            WILL BE INCURRED FOR EACH ADDITIONAL RUN.
            THIS IS RUN  2 OF DP-4.
   2. INPUT SELECTION MENU.
   3. SUBMIT A CONTRACT PROPOSAL
   4. EXIT
====>4
   PUN FILE 8029  TO  0276P    COPY 001   NOHOLD
   PUN FILE 8030  TO  0276P    COPY 001   NOHOLD
               ANOTHER RUN, 'Y'/'N'?
====>n
   END RECORDING OF TERMINAL SESSION
```

APPENDIX G

PROJMNG FORTRAN

```
C*************************************************************************
C
C                   P R O J M N G   F O R T R A N
C
C          NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA 93943
C
C                           MARCH 1984
C
C*************************************************************************
C
C                  Defense Management Simulation
C
C                Simulation and Computer Directorate
C               Industrial College of the Armed Forces
C               Fort Lesley J. McNair, Washington, D.C
C
C                          1970 April 1
C*************************************************************************
C
C The following fortran code listing represents the reformatted and
C aligned version of PROJMNG FORTRAN for inclusion as Appendix G
C in a thesis of the same name. The thesis PROJMNG FORTRAN was under-
C taken in June 1983 to upgrade the computer software support for the
C DMS exercise. The main effort of the thesis was to make the program
C user-friendly.
C
C Requests for this document should be referred to the Superintendent,
C Naval Postgraduate School, Monterey, California 93943.
C
C
C For a list of the variables definitions, see the Glossary,
C Appendix H
C
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &   ,     IFAC(48),    NT(48),     IRC(49),     IDDT(48)
     &   ,     IFD(48),   ITERM(48),    QTY(48),     CMAT(48)
      COMMON    RESS(48),    RESU(48),    RESB(48),     ITYPE(48)
     &   ,     PERC(48),    TDUE(48),    DEC(3),      Y(8)
     &   ,     IPRC(48),    RESC(48),    YF(48),      RESA(48)
```

```
    * ./,2X,'***CAUTION***IN RESPONSE TO A QUERY, PRESSING <<ENTER>>'
    *,/,6X,'WITHOUT PROVIDING DATA WILL DUMP THE PROGRAM.',/,1X,57('*')
    *,/,2X,'PAUSE;PRESS <<ENTER>> WHEN YOU ARE READY TO CONTINUE.',/,
    *1X,57('*'))
C...........................................................
C             M A I N   P R O G R A M   L O O P
C             ...........................
20    CONTINUE
      CALL ERRSET(218,1,-1,1,1)
C
C                                      (ESTABLISH INPUT VALUES)
C
                                   ITEAM=0
      CALL TEAMIN(ITEAM,MR,DATFIL)
10                         REWIND DATFIL
                           DO 25 JA=3,5
      READ(DATFIL,35,ERR=20,END=20) (NDUMP(JA,K),K=1,5),
     *(ADUMP(JA,K),K=1,28),STAT(JA)
35    FORMAT(2I2,1X,3I1,2X,8F7.2,/,3X,9F7.2,/,3X,6F7.2,/,3X,5F7.2,2X,A8)
25                         CONTINUE
      KDP=3
      IF(STAT(3).EQ.FINAL)KDP=4
      IF(STAT(4).EQ.FINAL)KDP=5
      IF(STAT(5).EQ.FINAL)KDP=6
      MAXDP=KDP
C
C                                 (INITIALIZE THE TEAM DATABASE)
C
C                                 (MONITOR SELECT THE DP- LEVEL)
                               IF(MR.EQ.0)GOTO 33
30                             PRINT 32,ITEAM,KDP,STAT(KDP)
32    FORMAT(1X,'TEAM ',I2,' IS AT DP-',I1,1X,A8,'.',/,
     *1X,'WHAT DECISION POINT DO YOU WISH TO ANALYZE?')
                               READ(IN,*,ERR=20,END=20)KDP
                               IF(KDP.GT.5.OR.KDP.LT.3)GOTO 30
                       IF(MR.EQ.3.OR.MR.EQ.2)GOTO 33
                                GOTO 410
C                                          (STUDENT DP- LEVEL)
33                         IF(NDUMP(3,2).NE.0)GOTO 410
      CALL PAGEIN
```

```
&    ,     TSTAR(48),    CL(6,5),    ALN(48),    CT(48)
&    ,     TFIN(48),    TMIN(48),    IQ(48),    CUN(30)
 COMMCN     RESMIN(30),     RESMAX(30),     AK(30),     ISW(3)
&    ,     ITAB(25),    NN(8),    N(8),    INDX(8)
&    ,     RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
&    ,     PER(6),    PVR(6),    CD,    ICD,    INA,INF,IDP
 COMMCN CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
&TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
 CCMMON     COSTP2,    COSTOT,    ITCT,    ALPHA,    PRICR
&    ,     TRADEP,    TF,    WA,    IACT(48,2),    IPC(48,5)
&    ,     TDS(48,2),    JMIN,    IC,    IFIG,    KDP,    MR
&    ,     NC,ICOFG,KP,KKPP,ITEAM
 COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
&FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
      CCMMON/PLOTVA/XBASE,YCURVE
      CCMMON/CODE/ ICODE,FLAG,ITEM,STORFL
      CCMMON/STORED/ STAT(5),NDUMP(5,5),ADUMP(5,28),DPCOST,MAXDP
      CCMMON/PAGES/ IPG(4)
      INTEGER ANS,ANO/'N'/,YES/'Y'/,E/'E'/,IN/5/,DATFIL,ANT,SUB/'S'/
      INTEGER CKPIT
      REAL*8 STAT,FINAL/'FINAL'/,INTEN/'PROPOSED'/
      REAL*8 ICODE,FLAG,ITEM,FL21/'TEAM21'/
      DIMENSION XBASE(11),YCURVE(11,11),IXBASE(11)
C...............................................................
C                        I N I T I A L I Z E
C THE FOLLOWING LINE IMPEDES STUDENT USE OF INSTRUCTOR OPTIONS; MR=1 OR
C2.
C
                  MR=0
                        DO 300 I=1,4
300               IPG(I)=I
                  DATFIL=9
                  KDP=3
C...............................................................
C     G E N E R A L   W A R N I N G S   A N D   I N S T R U C T I O N S
C     ..........................................................
5                       CALL FRTCMS('CLRSCRN ')
                        PRINT 11
11    FORMAT(58('*'),/,10X,'TO TERMINATE THE PROGRAM AT ANY POINT,',/,7X
     *,'TYPE  "E"=EXIT IN RESPONSE TO ANY "YES/NO" QUERY.',/,1X,57('*')
```

```
                                   CALL SETUP(KDP)
      CALL SELECT(KDP)
              CALL STORE
              GOTO 207
410                                CONTINUE
C
C                                   (UPDATE DATABASE BY 'SELECTION' MENU)
C                                     (START FROM LAST COMPLETE DATABASE)
C
                        JDP=KDP
75                                CONTINUE
                               IF(NDUMP(JDP,2).NE.0.OR.JDP.LE.3)GOTO 76
                        JDP=KDP-1
                               GOTO 75
76                                CONTINUE
      CALL SELECT(JDP)
C..............................................................................
C         P R E C A L C U L A T I O N   D A T A   P R O C E S S I N G
C         ....................................................................
C                                 (CHECK THAT FEE TOTAL PERCENTAGE IS 15%)
C
207   CALL PRCTCK
                                   CALL SETUP(KDP)
C
C     (STUDENT SPECIFIC PROCESSING TO PROVIDE FURTHER DATA PREPROCESSING
C     AND PREVENT UNDC WASTE OF CALCULATION TURN; AFTER 10 TURNS AT A DP
C     COST OF $100,000 PROCESSING FEE IS ASSESSED.)
C
230   CONTINUE
C
C          (DETERMINE DP, DECISION POINT;AND, PROMPT REPORT SUBMISSIONS)
C
                   IF(MR.NE.0)GOTO 231
      KDP=3
      IF(STAT(3).EQ.FINAL)KDP=4
      IF(STAT(4).EQ.FINAL)KDP=5
      IF(STAT(5).EQ.FINAL.AND.MR.EQ.0)STOP
      IF(STAT(5).EQ.FINAL.AND.MR.NE.0)GOTO 859
231                     NEWDP=KDP
                                      CALL SETUP(KDP)
```

```
                            CALL GET(KDP)
                   IF(KDP.NE.3)CALL INPUT2
C
                           DPCOST=0.0
                       DO 876 IKDP=3,KDP
876                        IF(ADUMP(IKDP,28).GT.10.)
    *           DPCOST=DPCOST+(ADUMP(IKDP,28)-10.)*100000.
                           IDUMP=ADUMP(KDP,28)+1.
C                      (BUFFER DATA ORGANIZATION FOR CALCULATIONS)
                   IF(MR.NE.0)GOTO 415
                            CALL STORE
C
875                        CALL FRTCMS('CLRSCRN ')
                                   IQUESE=4
                                   IQUESM=999
                           PRINT 211,KDP,IDUMP,KDP
211  FORMAT(' DO YOU WISH TO:',/,1X,'1. RECEIVE THE TABLE OF ACHIEVED V
    *ALUES.',/,10X,'*****WARNING*****AFTER THE TENTH DP-',I1,' RUN',/
    *,10X,'AN ADDITIONAL DEVELOPMENT COST OF $100,000',
    */,10X,'WILL BE INCURRED FOR EACH ADDITIONAL RUN.',/,10X,
    *'THIS IS RUN ',I2,' OF DP-',I1,'.')
                           IF(DPCOST.GT.0.)PRINT 225,DPCOST
225             FORMAT(10X,'DP RUNS HAVE COST AN ADDITIONAL $',F8.0)
                           PRINT 212
212  FORMAT(1X,'2. INPUT SELECTION MENU.')
                           PRINT 213
213  FORMAT(1X,'3. SUBMIT A CONTRACT PROPOSAL')
                           IF(MR.EQ.0)GOTO 216
                                   IQUESM=4
                                   IQUESE=5
                           PRINT 214,IQUESM
214  FORMAT(1X,I1,'. MONITOR''S MENU')
216                        PRINT 217,IQUESE
217  FORMAT(1X,I1,'. EXIT')
                           READ(IN,208,ERR=240,END=240)ANT
208                        FORMAT(I1)
                           IF(ANT.EQ.E)CALL EXITS
     IF(ANT.NE.1)GOTO 226
     IF(STORFL.EQ.999.)GOTO 415
C                          (ADD THE COST FOR THIS PRINTOUT)
```

```
                        IF(MR.EQ.0)ADUMP(KDP,28)=ADUMP(KDP,28)+1.
                            IF(ADUMP(IKDP,28).GT.10.)
                                DPCOST=DPCOST+100000.
     *
      GOTO 415
226   IF(ANT.EQ.IQUESM) GOTO 859
      IF(ANT.EQ.IQUESE) STOP
      IF(ANT.EQ.3) CALL FINISH
      IF(ANT.EQ.2) GOTO 25
      GCTO 230
240   CCNTINUE
      CALL EXITS
      GOTO 230
C
C...................................................................
C...................................................................
C         M A I N   P R O G R A M   C A L C U L A T I O N S
C
415       ----------CALL ZRW---------------------------------
                            NTEMP(1)=KDP
      DO 850 IVER=3,KDP
                    CALL GET(IVER)
80        CALL INPUT3
          CALL PRESET
          CALL TRADE
          CALL DESRES
                            GOTO(90,100),KKPP
90        CALL DESRET
100       CALL SE
      IF(NTEMP(1).EQ.3.OR.NTEMP(1).EQ.IDP) GOTO 130
                    ITCT=0
                            DO 110 M=1,INA
                               IF (QTY(M)-0.1) 110,110,115
115                            IF(ITERM(M)-IDP-1) 110,120,110
120                            ITCT=ITCT+1
110                         CONTINUE
                           GO TO 140
130                 IDP=7
                    ITCT=-1
                    ITAB(25)=IDP
C
```

```
140         CALL MOD9
                                  IF(IVER.EQ.KDP)GOTC 150
C                                 IF(IVER.NE.KDP-1)GCTO 850
                        CALL RERUN
850   CCNTINUE
150         CALL REPCRT
            CALL PERF
C                                (PRINTOUT THE TABLE OF ACHIEVED VALUES)
                        CALL GET(KDP)
            CALL PROSUM
C                                    (PREPARE BUFFER DATA FOR RELCOPING)
C.....................................................................
C.....................................................................
                        KDP=NEWDP
161                               IF(MR.NE.0)GOTO 859
      GOTO 230
859                               CALL FRTCMS('CLRSCRN ')
                         IQUESE=999
                         IQUESO=999
                         IQUESP=999
                         IQUESS=999
                         IQUESM=999
                         IQUSPP=999
                         IQUEPP=999
                         IQUMR2=999
                                  CALL GET(KDP)
                                  PRINT 860
860          FORMAT(' MAIN MENU:',/,2X,'1. CHANGE TEAM NUMBER',/,2X,
     *'2. THE DP SELECTION QUERY',/,2X,
     *'3. THE INPUT SELECTICN MENU',/,2X,
     *'4. RERUN THE DATA CALCULATION')
                                  IF(DPCOST.GT.0.)PRINT 225,DPCOST
                                  PRINT 861
861   FORMAT(    2X,'5. DO A SENSITIVITY ANALYSIS',/,2X,'6. PRINTCUT CCS
     *T+FACTORS')
                                            IQUESE=7
                                       IF(MR.EQ.1)GOTO 862
                                            IQUMR2=IQUESE
                                            IQUESE=IQUMR2+1
                                                   PRINT 863,IQUMR2
```

```
863   FORMAT(1X,I2,'. FILE TEAM PROPOSAL')
862                               CKPLT=0.
                                  CALL PLOT(CKPLT)
                      IF(CKPLT.EQ.0)GOTO 885
                                  PCTFLG=1.0
                      IQUESS=IQUESE
                      IQUESP=IQUESS+1
                      IQUEPP=IQUESP+1
                      IQUSPP=IQUEPP+1
                      IQUESO=IQUSPP+1
                      IQUESE=IQUESO+1
      PRINT 880,IQUESS,IQUESP,IQUEPP,IQUSPP,IQUESO,KDP
880   FORMAT(1X,I2,'. PRINT LAST SENSITIVITY TABLE.',/,1X,
     *I2,'. PLOT LAST SENSITIVITY ANALYSIS ACHIEVED VALUES.',/,1X,
     *I2,'. PLOT LAST SENSITIVITY ANALYSIS FEE PERCENTAGES.',/,1X,
     *I2,'. PLOT WITH SELECTED Y-AXIS SCALES.',/,1X,
     *I2,'. USE OPTIMUM EI VALUE FROM LAST SENSTIVITY ANALYSS FOR DP-',I
     *1,' DATA.')
885                   PRINT 890,IQUESE
890   FORMAT(1X,I2,'. EXIT')
              READ(IN,*,ERR=870,END=870)ANS
                           REWIND IN
                  IF(MR.NE.2)GOTO 877
                  IF(ANS.NE.IQUESE.AND.ANS.NE.1)GOTO 877
                          PRINT 27,ITEAM
27    FORMAT(1X,'SHOULD THE REWRITTEN TEAM ',I2,' FILE BE SENT TO THE PR
     *CMNG DISK: Y/N ?')
                          READ(IN,28,ERR=870,END=870)ANT
28                                FORMAT(A1)
                  IF(ANT.NE.YES)GOTO 877
      CALL SETUP(KDP)
      CALL STORE
      CALL FRTCMS('CP       ','SPOOL   ','PUN      ','CONT     ','CL      '
     *,'X        ','TC       ','0276P   ')
      CALL FRTCMS('PUN      ','DATAFILE',ITEM)
      CALL FRTCMS('CP       ','SPOOL   ','PUN      ','CLOSE    ','NOCONT  '
     *)
      IF(ANS.EQ.1)MR=1
877   CONTINUE
                      IF(ANS.EQ.IQUESE)STOP
```

```
      IF(ANS.EQ.1.OR.ANS.EQ.2.OR.ANS.EQ.3.)CKPLT=0.0
      IF(ANS.EQ.1.OR.ANS.EQ.2.OR.ANS.EQ.3.)
     *CAIL PLTNOT(CKPIT)
      IF(ANS.EQ.1)STCRFL=0.0
      IF(ANS.EQ.1)GOTC 20
      IF(ANS.EQ.2)GOTC 30
      IF(ANS.EQ.3)GOTC 75
      IF(ANS.EQ.4)GOTO 415
      IF(ANS.EQ.5)CAII SNSITV
      IF(ANS.EQ.6)CAII PROSM2
      IF(ANS.EQ.IQUESS)CALL SNSPRT
      IF(ANS.EQ.IQUESP)CALL PLTSCH
      IF(ANS.EQ.IQUEPP)CALL PLTPCT(PCTFLG)
      IF(ANS.NE.IQUSPP)GOTO 891
                                            PRINT 894
894   FORMAT(1X,'LOWEST VALUE FOR THE Y-AXIS')
                                            READ(IN,*,ERR=859,END=859)YMN
                                            PRINT 892
892   FORMAT(1X,'HIGHEST VALUE FOR THE Y-AXIS')
                                            READ(IN,*,ERR=859,END=859)YMX
      CAIL PLTSCL(YMN,YMX)
891   IF(ANS.EQ.IQUESC)CALL OPTIM
      IF(ANS.EQ.IQUESC)GOTO 75
      IF(ANS.EQ.IQUMR2)CALL FINISH
      IF(MR.EQ.2)CALI STORE
      IF(MR.EQ.2)CALI SETUP(KDP)
      GOTO 859
870                                     CALL EXITS
      GOTO 20
      END
C*********************************************************************
      SUBROUTINE EXITS
C
C THIS SUBROUTINE PROVIDES A COMMON EXITING ROUTINE FROM ANY YES/NO
C QUERY IN THE PROGRAM. IT CONSIDERS THE NEED TO CLOSE ANY RECORD SES-
C SIONS IN EFFECT. ALL DATABASE VARIABLES MUST BE ACCESSIBLE TO 'FINISH'
C THROUGH THIS SUBROUTINE.
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &   ,   IFAC(48),   NT(48),   IRC(49),   IDDT(48)
```

```
      &   ,   IFD(48),    ITERM(48),    QTY(48),    CMAT(48)
       CCMMON  RESS(48),    RESU(48),    RESB(48),    ITYPE(48)
      &   ,   PERC(48),   TDUE(48),    DEC(3),    Y(8)
      &   ;   IPRC(48),   RESC(48),    YF(48),    RESA(48)
      &   ;   TSTAR(48),   CL(6,5),    ALN(48),    CT(48)
      &   ,   TFIN(48),   TMIN(48),    IQ(48),    CUN(30)
       CCEMCN  RESMIN(30),    RESMAX(30),    AK(30),    ISW(3)
      &   ,   ITAB(25),    NN(8),    N(8),    INDX(8)
      &   ;   RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
      &   ,   PER(6),   PVR(6),    CD,    ICD,    INA,INF,IDF
       COMMCN CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
      &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
       COMMON  COSTP2,   COSTOT,    ITCT,    ALPHA,    PRICR
      &   ,   TRADEP,   TF,    WA,    IACT(48,2),    IPC(48,5)
      &   ;   TDS(48,2),    JMIN,    IC,    IFIG,    KDP,    MR
      &   ,   NC,ICOFG,KP,KKPP,ITEAM
       COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
      &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
              COMMON/CODE/ ICODE,FLAG,ITEM,STORFL
              REAL*8 ICODE,FLAG,ITEM,FL21/'TEAM21'/
              INTEGER ANS,ANO,YES,E,IN/5/,S
              COMMON/STORED/ STAT(5),NDUMP(5,5),ADUMP(5,28),DPCOST,MAXDP
              REAL*8 STAT,USERID
              REAL*8 FINAL/'FINAL'/,INTEN/'PROPOSED'/
              DATA ANC/'N'/,YES/'Y'/,E/'E'/,S/'S'/
       REWIND IN
10                               PRINT 15
15            FORMAT(1X,'DO YOU WISH TO CONTINUE;"Y"=YES CR "N"=NO?')
                                 READ(IN,25,ERR=10,END=10)ANS
25                               FORMAT(A1)
                                      REWIND IN
                         IF(MR.NE.2)GOTO 877
                         IF(ANS.EQ.YES)GOTO 877
                                     PRINT 27,ITEAM
27     FORMAT(1X,'SHOULD THE REWRITTEN TEAM ',I2,' FILE BE SENT TO THE PR
      *OMNG DISK: Y/N ?')
                                     READ(IN,28,ERR=10,END=10)ANT
28                                       FORMAT(A1)
                         IF(ANT.NE.YES)GOTO 877
       CALL SETUP(KDP)
```

```
      CALL STORE
      CALL FRTCMS('CP        ','SPOOL   ','PUN     ','CONT    ','CL      '
     *,'X       ','TC        ','0276P   ')
      CALL FRTCMS('CP        ','PUN     ','DATAFILE',ITEM)
      CALL FRTCMS('CP        ','SPOOL   ','PUN     ','CLOSE   ','NOCONT  '
     *)
877   CONTINUE
         IF(ANS.EQ.E)GOTO 999
         IF(ANS.EQ.YES)RETURN
         IF(ANS.EQ.ANC)GOTO 999
         IF(ANS.EQ.S)CALL FINISH
         IF(ANS.EQ.YES)RETURN
         GOTO 10
C.......................................................................
C PROGRAM TERMINATION BRANCHES
C                                         (DETERMINE IF STUDENT OR INSTRUCTOR)
999                                     CONTINUE
                                        IF(MR.EQ.1)GOTO 55
C                                                     (STUDENT TERMINATION)
C                                          (DETERMINE IF FILES WERE GENERATED)
50       CALL STORE
C                                     (STORED NEW OR REVISED DATA FOR STUDENTS)
      STOP
C.......................................................................
C                                                  (INSTRUCTOR TERMINATION)
55            CONTINUE
      STOP
      END
C **********************************************************************
      SUBROUTINE GET(JDP)
C
C IN ORDER TO FORMAT THE DATABASE BUFFER, THIS ROUTINE PUTS THE INPUT
C VARIABLES INTO AN ARRAY.
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &    ,    IFAC(48),    NI(48),    IRC(49),    IDDT(48)
     &    ,    IFD(48),   ITERM(48),   QTY(48),   CMAT(48)
      COMMON    RESS(48),    RESU(48),    RESB(48),    ITYPE(48)
     &    ,    PERC(48),    TDUE(48),    DEC(3),    Y(8)
     &    ,    IPRC(48),    RESC(48),    YF(48),    RESA(48)
```

```
&     ,     TSTAR(48),    CL(6,5),    ALN(48),    CT(48)
&     ,     TFIN(48),   TMIN(48),    IQ(48),    CUN(30)
 COMMON    RESMIN(30),    RESMAX(30),    AK(30),    ISW(3)
&     ,     ITAB(25),    NN(8),    N(8),    INDX(8)
&     ,     RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
&     ,     PER(6),    PVR(6),    CD,    ICD,    INA,INF,IDP
 COMMON CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV
&     ,     C10,    CO(18)
&     ,     IACT(48,2),    IPC(48,5)
&     ,     TDS(48,2),    JMIN,    IC,    IFIG,    KDP,    MF
 COMMON  NC,ICOFG,KP,KKPP,ITEAM
 COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
&FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
C
        COMMON/STORED/ STAT(5),NDUMP(5,5),ADUMP(5,28),DPCOST,MAXDP
        REAL*8 STAT
25         NC=NDUMP(JDP,3)
        ICOFG=NDUMP(JDP,4)
       MIS(6)=NDUMP(JDP,5)
       CTC(8)=ADUMP(JDP,1)
       CTC(7)=ADUMP(JDP,2)
       CTC(5)=ADUMP(JDP,3)
        YC(1)=ADUMP(JDP,4 )
        YC(2)=ADUMP(JDP,5 )
        YC(3)=ADUMP(JDP,6 )
        YC(4)=ADUMP(JDP,7 )
        YC(5)=ADUMP(JDP,8 )
        QT(2)=ADUMP(JDP,9 )
        QT(3)=ADUMP(JDP,10)
        QT(6)=ADUMP(JDP,11)
        QT(4)=ADUMP(JDP,12)
        QT(5)=ADUMP(JDP,13)
        QT(1)=ADUMP(JDP,14)
        CDMAX=ADUMP(JDP,15)
        CDMIN=ADUMP(JDP,16)
       FCDMIN=ADUMP(JDP,17)
        TDMAX=ADUMP(JDP,18)
        TDMIN=ADUMP(JDP,19)
       FTDMIN=ADUMP(JDP,20)
        RSMAX=ADUMP(JDP,21)
```

```
               RSMIN=ADUMP(JDP,22)
              FRSMAX=ADUMP(JDP,23)
               ESMAX=ADUMP(JDP,24)
               ESMIN=ADUMP(JDP,25)
              FESMIN=ADUMP(JDP,26)
              DD(13)=ADUMP(JDP,27)
99      RETURN
        END
C ***********************************************************************
        SUBROUTINE SELECT(JDP)
C
C THE MENU GENERATED PROVIDES BOTH A TABLE TO VERIFY THE CURRENT VALUES
C OF INPUT VARIABLES, AND ENABLES CHANGE OF THE DATA BY EITHER TOTAL
C LIST OF DATA, BY PAGE OF DATA, OR BY INDIVIDUAL VARIABLE. THE DATA
C INPUTS ARE ALSO CHECKED FOR VALIDITY, WITHIN DESIGN RANGE;AND, ASSISTS
C UPDATES BY PROMPTING RELATED GROUPS OF DATA AND BY SPECIFYING GAME
C RANGES.
C
        COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
       &   ,    IFAC(48),   NT(48),    IRC(49),    IDDT(48)
       &   ,    IFD(48),   ITERM(48),   QTY(48),    CMAT(48)
        COMMON    RESS(48),   RESU(48),    RESB(48),    ITYPE(48)
       &   ,    PERC(48),   TDUE(48),    DEC(3),    Y(8)
       &   ,    IPRC(48),   RESC(48),    YF(48),    RESA(48)
       &   ,    TSTAR(48),   CL(6,5),    ALN(48),    CT(48)
       &   ,    TFIN(48),   TMIN(48),    IQ(48),    CUN(30)
        COMMON    RESMIN(30),   RESMAX(30),    AK(30),    ISW(3)
       &   ,    ITAB(25),   NN(8),   N(8),    INDX(8)
       &   ,    RATIO(8),   AN(9),   ADJ(2,8),    PDR(6)
       &   ,    PER(6),   PVR(6),   CD,   ICD,   INA,INF,IDP
        COMMCN CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV
       &   ,    C10,   CQ(18)
       &   ,    IACT(48,2),    IPC(48,5)
       &   ,    TDS(48,2),   JMIN,   IC,   IFIG,   KDP,   MR
        COMMON  NC,ICOFG,KP,KKPP,ITEAM
        COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
       &FCDMIN,TDMAX,TIMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
C
              COMMON/STORED/ STAT(5),NDUMP(5,5),ADUMP(5,28),DECOST,MAXDP
                 COMMCN/PAGES/ IPG(4)
```

```
               REAL*8 STAT
               INTEGER ANS,IN/5/,E/'E'/,YES/'Y'/,ANO/'N'/
                                             CALL ERRSET(218,1,-1,1,1)
               CALL GET(JDP)
10             CALL FRTCMS('CLRSCRN ')
20             CALL SETUP(KDP)
15                                       PRINT 14,ITEAM,MAXDP,STAT(MAXDP)
14      FORMAT(15X,'*********TEAM ',I2,' IS IN DP-',I1,1X,A8,'.*********')
        PRINT 25,KDP,NDUMP(KDP,3)
     *           ,ADUMP(KDP,13),NDUMP(KDP,4),ADUMP(
     *KDP,14),NDUMP(KDP,5),ADUMP(KDP,15),ADUMP(KDP,1),
     *          ADUMP(KDP,16)
        PRINT 30,ADUMP(KDP,2),ADUMP(KDP,17),
     )          ADUMP(KDP,3),ADUMP(KDP,18),
     )          ADUMP(KDP,4),ADUMP(KDP,19),
     )          ADUMP(KDP,5),ADUMP(KDP,20),
     )          ADUMP(KDP,6),ADUMP(KDP,21),
     )          ADUMP(KDP,7),ADUMP(KDP,22)
        PRINT 35,ADUMP(KDP,8),ADUMP(KDP,23),
     )          ADUMP(KDP,9),ADUMP(KDP,24),
     )          ADUMP(KDP,10),ADUMP(KDP,25),
     )          ADUMP(KDP,11),ADUMP(KDP,26),
     )          ADUMP(KDP,12),ADUMP(KDP,27)
                IF(FLAG.EQ.2.)RETURN
25      FORMAT(' NO.*ITEM******CURRENT VALUE @ DP-',I1,4X,'NO.*ITEM*******
     ******CURRENT VALUE',/,'  1. CONTRACTOR                   :',2X,I1,8X,'16.
     * GUIDANCE QUAL TESTS    :',F6.2,/,'  2. GUIDANCE APPROACH       : ',1
     *X,I1,8X,'17. FLIGHT TESTS',10X,':',F6.2,/, '  3. GUIDANCE CONFIGUR
     *ATION:',2X,I1,8X, '18. MAXIMUM COST           M$:',F6.2,/,'  4. MAINT
     *AINABILITY ENG M$:',F6.2,  5X,'19. MINIMUM COST           M$:',F6.2)
30      FORMAT('  5. VALUE ENG            M$:',F6.2,4X,
     *          ' 20. MAX COST INCENTIVE   %:',F6.2
     *          ,/,'  6. PARALLEL DEVELOP    M$:',F6.2,4X,
     *          ' 21. LATEST WEEK          :',F6.2
     *          ,/,'  7. MOTOR RELIABILITY    %:',F6.2,4X,
     *          ' 22. EARLIEST WEEK        :',F6.2
     *          ,/,'  8. AIRFRAME RELIABILITY %:',F6.2,4X,
     *          ' 23. MAX DELIVER INCENTIVE%:',F6.2
     *          ,/,'  9. LAUNCHER RELIABILITY %:',F6.2,4X,
     *          ' 24. MAXIMUM RELIABILITY  %:',F6.2
```

```
     *              /,' 10. FIRE CONTROL ERROR YDS:',F6.2,4X,
     *              ' 25. MINIMUM RELIABILITY   %:',F6.2)
35    FORMAT(' 11. GUIDANCE ERROR       YDS:',F6.2,4X,
     *              ' 26. MAX RELIABIL INCENTIV%:',F6.2
     *              /,' 12. MOTOR QUAL TESTS      :',F6.2,4X,
     *              ' 27. MAXIMUM ERROR       YDS:',F6.2
     *              /,' 13. AIRFRAME QUAL TESTS   :',F6.2,4X,
     *              ' 28. MINIMUM ERROR       YDS:',F6.2
     *              /,' 14. LAUNCHER QUAL TESTS   :',F6.2,4X,
     *              ' 29. MAX ERROR INCENTIVE   %:',F6.2
     *              /,' 15. FIRE CONTROL QUAL TSTS:',F6.2,4X,
     *              ' 30. WEEK FOR LOT 10        :',F6.2)
C.....................................................................
C        C H A N G E   F O R M A T   S E L E C T I O N
C
132                             PRINT 40
C                                      (MENU EXPANSION FOR CHANGE SIZING)
40    FORMAT(15X,' *****0. NONE**********31. CHANGE BY PAGE*****')
                                PRINT 24,KDP
24    FORMAT(13X,' SELECT AN ITEM WHICH NEEDS TO BE CHANGED FOR DP-',I1
     *          ,'.')
            GOTO 130
41    CONTINUE
C                                             (ZEROIZE PAGE FLAGGING)
                                DO 4025 I=1,4
4025                            IPG(I)=0
                                CALL FRTCMS('CLRSCRN ')
                                PRINT 4015
4015  FORMAT(' *****TYPE IN A LIST OF PAGES YOU DESIRE TO CHANGE:*****'/
     1' EX:1234 WILL ASK FOR NEW DATA ON ALL PAGES; OR, 23 WILL ASK FOR'
     2/' NEW DATA ON PAGES 2 AND 3.',/,' *****WARNING***** YOU MUST ENTE
     3R ATLEAST ONE DIGIT EVEN IF IT IS A 0.')
                               READ(IN,4020,ERR=41,END=41)(IPG(I),I=1,4)
4020                           FORMAT(4I1)
      CALL PAGEIN
      GOTO 10
C.....................................................................
C    R E T U R N   T O   M A I N   P R O G R A M   P R O C E S S I N G
C
130         CONTINUE
```

```
                                        READ(IN,*,ERR=133,END=133)ANS
                IF(ANS.EQ.0)RETURN
                IF(ANS.GE.1.OR.ANS.LT.32)GO TO 160
133                                     CALL EXITS
      RETURN
C.........................................................................
C           D A T A   I N P U T   P R O C E S S I N G
C
160   CONTINUE
                                        IANS=ANS
      IF(KDP.GE.4.AND.(ANS.EQ.1.OR.ANS.EQ.2))
     *                                  CALL FRTCMS('CLRSCRN ')
                                        IF(KDP.GE.4.AND.(ANS.EQ.1.OR.ANS.EQ.2))
     *                                  PRINT 162
162   FORMAT(' AT DP-4, CONTRACTOR AND GUIDANCE APPROACH ARE FROZEN.',
     */,' AND, GUIDANCE CONFIGURATION MUST BE SOLELY EITHER "A" OR "B".'
     *,24(/))
                                        IF(KDP.GE.4.AND.(ANS.EQ.1.OR.ANS.EQ.2))
     *GOTO 10
      IF(ANS.LT.7.AND.KDP.GT.4)GOTO (160,160,401,500,600,701),IANS
                 GOTO (200,300,400,500,600,700,800,900,1000,1100,1200
     *,1300,1400,1500,1600,1700,1800,1900,2000,2100,2200,2300,2400,2500
     *,2600,2700,2800,2900,3000,3100,41,3200),IANS
150                                     CONTINUE
C
      GOTO 10
C
200   CALL CHANGI(NC)
210   IF ((NC.EQ.1) .OR. (NC.EQ.2)) GO TO 10
220   PRINT 230, NC
230   FORMAT (/,' CONTRACTOR=',I2,' IS INVALID; PLEASE RETYPE CONTRACTOR
     * NUMBER; EITHER "1" OR "2".')
      GO TO 200
C
300   CALL CHANGI(ICOFG)
305   IF ((ICOFG.EQ.1) .OR. (ICOFG.EQ.2)) GOTO 10
      PRINT 310, ICOFG
310   FORMAT (/,' GUIDANCE APPROACH=',I2,' IS INVALID; PLEASE RETYPE GUI
     *DANCE APPROACH; "1" OR "2".')
      GO TO 300
```

```
C
400    CAIL CHANGI(MIS(6))
401    IF(KDP.EQ.3)GOTC 405
       IF(ADUMP(KDP,5).GT.0.)CTC(5)=0.
       IF(MIS(6).EQ.2.CR.MIS(6).EQ.4)GOTO 405
       PRINT 131
131    FORMAT(' AT DP-4 FINAL GUIDANCE CONFIGURATION MUST EE SELECTED.'
      *,/,' ELEASE SELECT EITHER 2 OR 4 FOR GUIDANCE CONFIGURATION.')
       GOTO 400
405    IF ((MIS(6).GE.1) .AND. (MIS(6).LE.4)) GOTO 701
       PRINT 410, MIS(6)
410    FORMAT (/,' GUIDANCE CONFIGURATION=',I2,' IS INVALIC; RETYPE GUIDA
      *NCE CCNFIGURATICN;"1","2","3",OR "4".')
       GO TO 400
C
500    CAIL CHANGE(CTC(8))
       XC8 = 1.5
505    IF (CTC(8).GE.XC8) GOTO 10
       PRINT 510
510    FORMAT (1H0, 'MAINT')
       PRINT 515, CTC(8), XC8
515    FORMAT (1H+, ' ENG FUNDS OF $', F5.2,
      &   ' HAVE EEEN RESET TO THE $', F5.2, ' MINIMUM.')
       CTC(8) = XC8
520    PRINT 525
525    FORMAT(1H+,'MAINT ENG FUNDS = $ ',F5.2)
       GO TO 500
C
600    CAIL CHANGE(CTC(7))
       XC7 = 0.5
       IF (CTC(7).GE.XC7)  GOTO 10
       PRINT 610
610    FORMAT (1H0, 'VALUE')
       PRINT 515, CTC(7), XC7
       CTC(7) = XC7
       PRINT 620, CTC(7)
620    FORMAT(1H+,'VALUE ENG FUNDS = $',F5.2)
       GO TO 600
C
700    CAIL CHANGE(CTC(5))
```

```
701   IF(KDP.EQ.3)GOTO 704
      IF(CTC(5).EQ.0.)GOTO 10
      CTC(5)=0.
      PRINT 703
703   FORMAT(' PARALLEL DEVELOPMENT FUNDS ARE NOT AVAILABLE AFTER DP-3',
     */' PLEASE DO NOT ATTEMPT TO ADD PARALLEL DEVELOPMENT FUNDS; AND',/
     */,' DO NOT ATTEMPT TO CHANGE GUIDANCE CONFIGURATION TO PARALLEL CO
     *NFIGURATIONS.')
      PRINT 702
702   FORMAT(20(/))
      IF(MIS(6).EQ.1.OR.MIS(6).EQ.3)GOTO 401
      GOTO 10
704   IF ((MIS(6).EQ.1) .OR. (MIS(6).EQ.3)) GOTO 750
      IF (CTC(5).LE.0.0) GOTO 10
      PRINT 705
705   FORMAT (1H0, 'PARALLEL DEV FUNDS')
      PRINT 710,CTC(5)
710   FORMAT (1H+, ' OF $', F5.2,'M')
      PRINT 720
720   FORMAT (1H+, ' ARE NOT REQUIRED AND HAVE BEEN RESET TO $0.0')
      CTC(5) = 0.0
      PRINT 725
725   FORMAT(1X,'DO YOU WISH TO CHANGE GUIDANCE CONFIGURATION: "Y"=YES O
     *R "N"=NO?')
      READ(IN,730,ERR=704,END=704)ANS
730   FORMAT(A1)
        IF(ANS.EQ.E)CALL EXITS
        IF(ANS.EQ.ANO)GO TO 10
      GO TO 400
C
750   IF (CTC(5).GT.0.0) GOTO 10
      PRINT 705
      PRINT 755
755   FORMAT (1H+, ' ARE REQUIRED.')
      PRINT 760
760   FORMAT(1X,'DO YOU WISH TO CHANGE GUIDANCE CONFIGURATION: "Y"=YES O
     *R "N"=NO?')
      READ(IN,765,ERR=750,END=750)ANS
765   FORMAT(A1)
        IF(ANS.EQ.E)CALL EXITS
```

```
      IF(ANS.EQ.ANC)GO TO 700
      GOTO 400
C
800   CALL CHANGE(YC(1))
      IF(YC(1).GE.0..AND.YC(1).LE.100.0)GO TO 10
      PRINT 805,YC(1)
805   FORMAT(' RELIABILITY EXCEEDED 100%; RELIABILITY=',F3.1,'.',/,' PLE
     *ASE REENTER RELIABILITY.')
      GOTO 800
C
900   CALL CHANGE(YC(2))
      IF(YC(2).GE.0..AND.YC(2).LE.100.0)GO TO 10
      PRINT 805,YC(2)
      GOTO 900
C
1000   CALL CHANGE(YC(3))
      IF(YC(3).GE.0..AND.YC(3).LE.100.0)GO TO 10
      PRINT 805,YC(3)
      GOTO 1000
C
1100  CALL CHANGE(YC(4))
      IF(YC(4).GE.0..AND.YC(4).LE.1000.0)GO TO 10
      PRINT 1105,YC(4)
1105  FORMAT(' ERROR EXCEEDED 1000YDS; ERROR=',F3.1,'.',/,' PLEASE REENT
     *ER ERROR.')
      GOTO 1100
C
1200  CALL CHANGE(YC(5))
      IF(YC(5).GE.0..AND.YC(5).LE.1000.0)GO TO 10
      PRINT 1105,YC(5)
      GOTO 1200
C
1300  CALL CHANGE(QT(2))
      IF(QT(2).GE.20..AND.QT(2).LE.40.)GO TO 10
      PRINT 1305,QT(2)
1305  FORMAT(1X,'MOTOR QUALIFICATION TESTS MUST BE GREATER THAN 20 AND L
     *ESS THAN 40. YOUR QT(2) =',F2.0)
      GO TO 1300
C
1400  CALL CHANGE(QT(3))
```

```
      IF(QT(3).GE.3..AND.QT(3).LE.9.)GO TO 10
      PRINT 1405,QT(3)
1405  FORMAT(1X,'AIRFRAME QUALIFICATION TESTS MUST BE GREATER THAN 3 AND
     * LESS THAN 9. YCUR TESTS =',F1.0)
      GO TO 1400
C
1500           CALL CHANGE(CT(6))
      IF(CT(6).GE.2..AND.QT(6).LE.6.)GO TO 10
      PRINT 1505,QT(6)
1505  FORMAT(1X,'LAUNCHER QUALIFICATION TESTS MUST BE GREATER THAN 2 AND
     * LESS THAN 6. YCUR TESTS =',F1.0)
      GO TO 1500
C
1600           CALL CHANGE(QT(4))
      IF(QT(4).GE.2..AND.QT(4).LE.4.)GO TO 10
      PRINT 1615,QT(4)
1615  FORMAT(1X,'FIRE CONTROL QUALIFICATION TESTS MUST EE GREATER THAN 2
     * AND LESS THAN 4. YOUR TESTS =',F1.0)
      GO TO 1600
C
1700           CALL CHANGE(CT(5))
      IF(QT(5).GE.3..AND.QT(5).LE.9.)GO TO 10
      PRINT 1725,QT(5)
1725  FORMAT(1X,'GUIIANCE QUALIFICATION TESTS MUST BE GREATER THAN 3 AND
     *LESS THAN 9. YCUR TESTS =',F1.0)
      GC TO 1700
C
1800           CALL CHANGE(QT(1))
      IF(QT(1).GE.10..AND.QT(1).LE.25.)GO TO 10
      PRINT 1835,QT(1)
1835   FORMAT(1X,'FLIGHT TESTS MUST BE GREATER THAN 10 AND LESS THAN 25.
     *YOUR TESTS =',F2.0)
      GO TO 1800
C
1900  CAIL CHANGE(CDMAX)
      PRINT 1910
1910  FORMAT(' DO YOU DESIRE TO CHANGE MINIMUM DEVELCPMENI COST, ALSO.')
      READ(IN,2520,ERR=1900,END=1900)ANS
      IF(ANS.EQ.E)CAII EXITS
      IF(ANS.EQ.ANO)GCTO 10
```

```
2000            CALL CHANGE(CDMIN)
2010     IF(CDMAX.GE.CDMIN)GOTO 10
         PRINT 1995,CDMAX,CDMIN
1995     FORMAT(1X,'MAX COST IS LESS THAN MIN COST. MAX COST =',F4.1,
        *',MIN  COST=',F4.1)
                GO TO 1900
C
2100     CALL CHANGE(FCDMIN)
         GOTO 10
C
2200            CALL CHANGE(TDMAX)
         PRINT 2210
2210     FORMAT(' DO YOU DESIRE TO CHANGE MINIMUM FLIGHT TEST COMPLETION, A
        *LSO.')
         READ(IN,2520,ERR=2200,END=2200)ANS
         IF(ANS.EQ.E)CALL EXITS
         IF(ANS.EQ.ANO)GOTO 10
2300            CALL CHANGE(TDMIN)
2310     IF(TDMAX.GE.TDMIN)GOTO 10
         PRINT 2296,TDMAX,TDMIN
2296     FORMAT(1X,'FLIGHT TEST LATEST WEEK IS LESS THAN EARLIEST WEEK;'/
        *' LATEST DATE = ',F6.0,';EARLIEST DATE =',F6.0)
                GO TO 2200
C
2400     CALL CHANGE(FTDMIN)
         GOTO 10
C
2500     CALL CHANGE(RSMAX)
         PRINT 2510
2510     FORMAT(' DO YOU DESIRE TO CHANGE MINIMUM RELIABILITY, ALSO.')
         READ(IN,2520,ERR=2500,END=2500)ANS
2520     FORMAT(A1)
         IF(ANS.EQ.E)CALL EXITS
         IF(ANS.EQ.ANO)GOTO 10
2600     CALL CHANGE(RSMIN)
2610     IF(RSMAX.GE.RSMIN)GOTO 10
         PRINT 2505,RSMAX,RSMIN
2505     FORMAT(1X,'MAXIMUM SYSTEM RELIABILITY IS LESS THAN',/,5X,
        *'MINIMUM RELIABILITY: RSMAX =',F6.0,'; RSMIN =',F6.0)
         GOTO 2500
```

```
C
2700   CALL CHANGE(FRSMAX)
       GOTO 10
C
2800   CALL CHANGE(ESMAX)
       GOTO 10
C
2900   CALL CHANGE(ESMIN)
       GOTO 10
C
3000   CALL CHANGE(FESMIN)
       GOTO 10
C
3100   CALL CHANGE(DD(13))
       GOTO 10
C
3200   RETURN
       END
C**************************************************************************
       SUBROUTINE SETUP(QDP)
       COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &   ,     IFAC(48),   NI(48),    IRC(49),    IDDT(48)
     &   ,     IFD(48),   ITERM(48),    QTY(48),    CMAT(48)
       COMMON   RESS(48),    RESU(48),    RESB(48),    ITYPE(48)
     &   ,     PERC(48),    TDUE(48),    DEC(3),    Y(8)
     &   ,     IPRC(48),    RESC(48),    YF(48),    RESA(48)
     &   ,     TSTAR(48),    CL(6,5),    ALN(48),    CT(48)
     &   ,     TFIN(48),    TMIN(48),    IQ(48),    CUN(30)
       COMMON   RESMIN(30),    RESMAX(30),    AK(30),    ISW(3)
     &   ,     ITAB(25),    NN(8),    N(8),    INDX(8)
     &   ,     RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
     &   ,     PER(6),    PVR(6),    CD,    ICD,    INA,INF,IDP
       COMMON CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV
     &   ,     C10,    CQ(18)
     &   ,     IACT(48,2),    IPC(48,5)
     &   ,     TDS(48,2),    JMIN,    IC,    IFIG,    KDP,    MR
       COMMON  NC,ICOFG,KP,KKPP,ITEAM
       COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
C
```

```
      COMMON/STORED/ STAT(5),NDUMP(5,5),ADUMP(5,28),DPCCST,MAXDP
      REAL*8 STAT
      INTEGER DATFIL/9/,QDP
      NDUMP(QDP,1)=ITEAM
      NDUMP(QDP,2)=QDP
      NDUMP(QDP,3)=NC
      NDUMP(QDP,4)=ICOFG
      NDUMP(QDP,5)=MIS(6)
      ADUMP(QDP,1 )=CTC(8)
      ADUMP(QDP,2 )=CTC(7)
      ADUMP(QDP,3 )=CTC(5)
      ADUMP(QDP,4 )=YC(1)
      ADUMP(QDP,5 )=YC(2)
      ADUMP(QDP,6 )=YC(3)
      ADUMP(QDP,7 )=YC(4)
      ADUMP(QDP,8 )=YC(5)
      ADUMP(QDP,9 )=QT(2)
      ADUMP(QDP,10)=QT(3)
      ADUMP(QDP,11)=QT(6)
      ADUMP(QDP,12)=QT(4)
      ADUMP(QDP,13)=QT(5)
      ADUMP(QDP,14)=QT(1)
      ADUMP(QDP,15)=CDMAX
      ADUMP(QDP,16)=CDMIN
      ADUMP(QDP,17)=FCDMIN
      ADUMP(QDP,18)=TDMAX
      ADUMP(QDP,19)=TDMIN
      ADUMP(QDP,20)=FTDMIN
      ADUMP(QDP,21)=RSMAX
      ADUMP(QDP,22)=RSMIN
      ADUMP(QDP,23)=FRSMAX
      ADUMP(QDP,24)=ESMAX
      ADUMP(QDP,25)=ESMIN
      ADUMP(QDP,26)=FESMIN
      ADUMP(QDP,27)=DD(13)
      RETURN
      END
C ************************************************************************
      SUBROUTINE CHANGE(XXX)
C
```

```
      INTEGER IN/5/
      CALL ERRSET(218,1,-1, 1,1)
120                              PRINT 130,XXX
130   FORMAT(1X,'PREVIOUS VALUE ',F6.2,': INPUT A VALUE')
             READ(IN,*,ERR=150,END=150) XXX
      RETURN
150   CALL EXITS
      GO TO 120
      END
C ***************************************************************************
      SUBROUTINE CHANGI(IXX)
C
      INTEGER IN/5/
      CALL ERRSET(218,1,-1, 1,1)
120                              PRINT 130,IXX
130   FORMAT(1X,'PREVIOUS VALUE ',I3,': INPUT A VALUE')
             READ(IN,*,ERR=150,END=150) IXX
      RETURN
150   CALL EXITS
      GO TO 120
      END
C ***************************************************************************
C
      SUBROUTINE TEAMIN(ITEAM,MR,DATFIL)
C
      INTEGER ANS,ANC/'N'/,YES/'Y'/,E/'E'/,IN/5/,T/'M'/,DATFIL,IFAULT
      INTEGER DATCOD/11/,P/'P'/,ANT
      COMMON/STORED/ STAT(5),NDUMP(5,5),ADUMP(5,28),DPCCSI,MAXDP
      COMMON/CODE/ ICODE,FLAG,ITEM,STORFL
      REAL*8 STAT,USERID,ITEM,FLAG,FL21/'TEAM21  '/
      REAL*8 ICODE,INCOD,BLK/'        '/,ZERO/'00000000'/
      REAL*8 TEAM01/'TEAM01  '/,TEAM02/'TEAM02  '/,TEAM03/'TEAM03  '/,
     *TEAM05/'TEAM05  '/,TEAM06/'TEAM06  '/,TEAM07/'TEAM07  '/,TEAM08/'T
     *EAM08  '/,TEAM09/'TEAM09  '/,TEAM10/'TEAM10  '/,TEAM11/'TEAM11  '/
     *,TEAM12/'TEAM12  '/,TEAM13/'TEAM13  '/,TEAM14/'TEAM14  '/,TEAM15/
     *'TEAM15  '/,TEAM17/'TEAM17  '/,TEAM18/'TEAM18  '/,TEAM19/'TEAM19
     *'/,TEAM20/'TEAM20  '/,TEAM04/'TEAM04  '/,TEAM16/'TEAM16  '/
C...........................................(RESET PLOT FLAG TO ZERO)
                                          CALL ERRSET(218,1,-1,1,1)
                                          ISEN=0
```

```
                                      CALL PLOTST(ISEN)
C...................................................(GET THE TEAM NUMBER)
C                                              (CHECK THE MONITOR'S FLAG)
C                                                       (IF NO FILE EXIT)
                                  ITEAMB=999
                                  IF(FLAG.EQ.FL21.OR.MR.NE.0)GOTO 21
                                  READ(5,*)ITEAM
                                  READ(5,95)FLAG
                                  REWIND 9
                                  READ(9,16,ERR=17,END=17)ITEAMB
16                                FORMAT(I2)
17                                CONTINUE
      IF(ITEAM.EQ.21)GOTO 475
      IF(FLAG.EQ.FL21)GOTO 475
      GOTO 32
475                               ITEM=BLK
19                                PRINT 476
476   FORMAT(' PLEASE ENTER YOUR SECURITY CODE.')
                                  REWIND IN
                                  READ(IN,95,ERR=450,END=450)INCOD
                                  REWIND DATCOD
                                  READ(DATCOD,95,ERR=450,END=450)USERID
                                  READ(DATCOD,95,ERR=450,END=450)ICODE
                        IF(INCOD.NE.ICODE)GOTO 450
21                                PRINT 20
20    FORMAT(' SELECT WHICH INSTRUCTOR MODE YOU DESIRE.',/,' 1. RUN TEAM
     * SCENARIOS WHICH ARE ON THE MONITOR''S DISK.',/,' 2. CHANGE TEAM F
     *ILES.',/,8X,'THIS MODE WILL ALTER THE STUDENT''S FILES.')
                                  PRINT 23
23    FORMAT(' 3. RUN THE MONITOR''S TEAM FILE ''TEAM21''.',/,1X,'4. EXI
     *T')
      READ(IN,*,ERR=450,END=450)ANS
                                  IF(ANS.EQ.1)MR=1
                                  IF(ANS.EQ.2)MR=2
                                  IF(ANS.NE.3)GOTO 22
                                  MR=3
                                  ITEAM=21
           CALL FRTCMS('FILEDEF ','9       ','DISK    ','DATAFILE',FL21
     *                           ,'(       ','PERM    ')
                                  RETURN
```

```
22                              IF(ANS.NE.4)GOTO 877
                            IF(MR.NE.2) STOP
                                    PRINT 27,ITEAM
27    FORMAT(1X,'SHOULD THE REWRITTEN TEAM ',I2,' FILE BE SENT TO THE PR
     *CMNG DISK: Y/N ?')
                                    READ(IN,28,ERR=22,END=22)ANT
28                                      FORMAT(A1)
                            IF(ANT.NE.YES)GOTO 877
      CALL SETUP(KDP)
      CALL STORE
      CALL FRTCMS('CE      ','SPOOL   ','PUN     ','CONT    ','CL      '
     *,'X       ','TC      ','0276P   ')
      CALL FRTCMS('PUN     ','DATAFILE',ITEM)
      CALL FRTCMS('CE      ','SPOOL   ','PUN     ','CLOSE   ','NOCONT  '
     *)
      STOP
877   CONTINUE
                                IF(ITEM.EQ.BLK)GOTO 32
485                             CONTINUE
                                IF(ITEAM.EQ.21)GOTO 31
      IF(MR.EQ.1.AND.ITEAMB.EQ.ITEAM) RETURN
      IF(MR.EQ.2.AND.ITEAMB.EQ.ITEAM) RETURN
30                              CONTINUE
                IF(ITEAMB.NE.999.AND.ITEAMB.NE.ITEAM)PRINT 24,ITEAM
24    FORMAT(' RECORDS DO NOT EXIST FOR TEAM ',I2)
31                              PRINT 25
25    FORMAT(' WHAT TEAM NUMBER?')
      READ(IN,*,ERR=495,END=495) ITEAM
32                              CONTINUE
                                IF(ITEAM.LE.0.OR.ITEAM.GT.21)GOTO 485
                                IF(ITEAM.EQ.1) ITEM=TEAM01
                                IF(ITEAM.EQ.2) ITEM=TEAM02
                                IF(ITEAM.EQ.3) ITEM=TEAM03
                                IF(ITEAM.EQ.4) ITEM=TEAM04
                                IF(ITEAM.EQ.5) ITEM=TEAM05
                                IF(ITEAM.EQ.6) ITEM=TEAM06
                                IF(ITEAM.EQ.7) ITEM=TEAM07
                                IF(ITEAM.EQ.8) ITEM=TEAM08
                                IF(ITEAM.EQ.9) ITEM=TEAM09
                                IF(ITEAM.EQ.10) ITEM=TEAM10
```

```
                                  IF(ITEAM.EQ.11) ITEM=TEAM11
                                  IF(ITEAM.EQ.12) ITEM=TEAM12
                                  IF(ITEAM.EQ.13) ITEM=TEAM13
                                  IF(ITEAM.EQ.14) ITEM=TEAM14
                                  IF(ITEAM.EQ.15) ITEM=TEAM15
                                  IF(ITEAM.EQ.16) ITEM=TEAM16
                                  IF(ITEAM.EQ.17) ITEM=TEAM17
                                  IF(ITEAM.EQ.18) ITEM=TEAM18
                                  IF(ITEAM.EQ.19) ITEM=TEAM19
                                  IF(ITEAM.EQ.20) ITEM=TEAM20
                                  IF(ITEAM.EQ.21) ITEM=FI21
           CALL FRTCMS('FILEDEF ','9        ','DISK    ','DATAFILE',ITEM
     *                           ,'(        ','PERM    ')
                                  IF(MR.EQ.0)
     *CALL FRTCMS('FILEDEF ','11      ','DISK    ','DATACCDE',ITEM
     *                           ,'(        ','PERM    ')
                                  IF(MR.EQ.0) GOTO 15
                                  REWIND 9
                                  READ(9,16,ERR=485,END=485) ITEAME
                                  GOTO 485
15    CCNTINUE
                                  REWIND 9
                             READ(9,16,ERR=30,END=30)ITEAMB
                                  REWIND DATCOD
                             READ(DATCOD,95,ERR=450,END=450) ICODE
                             READ(DATCOD,230,ERR=450,END=450) IFAULT
                               IF(ICODE.EQ.BLK.OR.ICCDE.EQ.ZERO)GOTO 100
                                  IF(IFAULT.GE.5) GOTO 65
                                  GOTO 70
100                               PRINT 110
110    FORMAT('  *****YCU ARE ENCOURAGED TO CHANGE YCUR SECURITY CODE?****
     ****')
170                               PRINT 130
130           FORMAT(' ENTER YCUR NEW CODE. ')
                                  READ(IN,95,ERR=120,END=120) ICODE
              IF(ICOLE.EQ.BLK.OR.ICODE.EQ.ZERO)GOTO 120
                                  GOTO 99
120                                         CALL EXITS
                                                 GOTC 100
70                                PRINT 60
```

```
60             FORMAT(' PLEASE ENTER YOUR TEAM SECURITY CODE.')
                                 READ(IN,95,ERR=15,END=15) INCOD
              IF(INCOD.NE.ICODE) GO TO 200
99                               REWIND DATCOD
      IF(FLAG.NE.FL21)  WRITE(DATCOD,95) ICODE
95                               FORMAT(A8)
      IF(FLAG.NE.FL21)  WRITE(DATCOD,230)IFAULT
      RETURN
200                     IFAULT=IFAULT+1
                                 PRINT 210
210   FORMAT(' YOU HAVE ATTEMPTED TO ENTER A TEAM FILE WITHOUT THE PROPE
     *R SECURITY CODE.')
                                 REWIND DATCOD
      IF(FLAG.NE.FL21)  WRITE(DATCOD,95) ICODE
      IF(FLAG.NE.FL21)  WRITE(DATCOD,230) IFAULT
230                              FORMAT(I3)
                                 IF(IFAULT.LT.5)CALL EXITS
                                 IF(IFAULT.LT.5)GOTO 70
65    CONTINUE
                                 PRINT 310,ITEAM,IFAULT
310   FORMAT(' THE USER ID CODE FOR TEAM ',I2,' HAS BEEN TRIED INCORRECT
     *LY MORE THAN ',I3,' TIMES.',//,' PLEASE LOGOFF AND CONTACT YOUR MO
     *NITOR IN ORDER TO REGAIN ACCESS TO YOUR TEAM',/,' FILES.')
150   STOP
450   CONTINUE
                                 PRINT 455
455   FORMAT(1X,'YOUR INPUT WAS READ INCORRECTLY.',/,
     *' PLEASE REINITIATE THE PROGRAM TO CORRECT THE ERROR.')
                                 IF(FLAG.NE.FL21) PRINT 456
456   FORMAT(
     *' IF YOU HAVE FURTHER QUESTIONS PLEASE SEE YOUR CLASS'' MONITOR.')
      PAUSE ';PRESS <<ENTER>>'
495   STOP
      END
C
C*****************************************************************************
C
      SUBROUTINE PAGEIN
C
C THIS SUBROUTINE ENABLES MASS PAGE DATA INPUT.
```

```
C
      CCMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &   ,    IFAC(48),    NT(48),     IRC(49),    IDDT(48)
     &   ,    IFD(48),   ITERM(48),    QTY(48),    CMAT(48)
      CCMMON   RESS(48),    RESU(48),    RESB(48),    ITYPE(48)
     &   ,    PERC(48),    TDUE(48),    DEC(3),     Y(8)
     &   ,    IPRC(48),    RESC(48),    YF(48),     RESA(48)
     &   ,    TSTAR(48),    CL(6,5),    ALN(48),    CT(48)
     &   ,    TFIN(48),    TMIN(48),    IQ(48),     CUN(30)
      CCMMCN   RESMIN(30),    RESMAX(30),    AK(30),     ISW(3)
     &   ,    ITAB(25),    NN(8),    N(8),    INDX(8)
     &   ,    RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
     &   ,    PER(6),    PVR(6),    CD,    ICD,    INA,INF,IDP
      CCMMCN CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,IRADET,TRADER,
     &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
      CCMMCN   COSTP2,    COSTOT,    ITCT,    ALPHA,    PRICR
     &   ,    TRADEP,    TF,    WA,    IACT(48,2),    IPC(48,5)
     &   ,    TDS(48,2),    JMIN,    IC,    IFIG,    KDP,    MR
     &   ,    NC,ICOFG,KP,KKPP,ITEAM
      COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TIMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
      COMMON/PAGES/ IPG(4)
      INTEGER ANS,YES/'Y'/,ANO/'N'/,E/'E'/,IN/5/,QDP
      DIMENSION YTQ(6),ITQ(6)
C
      CAIL ERRSET(218,1,-1,1,1)
10                                     CALL FRTCMS('CLRSCRN ')
                                       DO 11 INV=1,4
      IF(IPG(INV).EQ.1) GOTO 15
11                                     CONTINUE
        GOTO 360
600                                    CALL EXITS
15                                     CONTINUE
25                                     PRINT 20,KDP
20    FORMAT(1X,'*****INPUT PAGE 1 DP-',I1,' INFO*****.'//)
                                       PRINT 55
55    FORMAT(' CONTRACTOR NUMBER,(1 OR 2):')
                                       READ (IN,*,ERR=600,END=600) NC
                                       PRINT 700
700   FORMAT(' GUIDANCE APPROACH,(1 OR 2):')
```

```
                                   READ (IN,*,ERR=600,END=600) ICOFG
                                   PRINT 705
705   FORMAT(' GUIDANCE CONFIGURATION,(1-4;NOT 1 OR 3 AFTER DE-3')
                                   READ(IN,*,ERR=600,END=600) MIS(6)
                                   PRINT 710
710   FORMAT(' MAINTAINABILITY ENGINEERING FUNDS IN $M:')
                                   READ(IN,*,ERR=600,END=600) CTC(8)
                                   PRINT 715
715   FORMAT(' VALUE ENGINEERING FUNDS IN $M:')
                                   READ(IN,*,ERR=600,END=600) CTC(7)
                        IF(MIS(6).EQ.2.OR.MIS(6).EQ.4) GO TO 80
                                   PRINT 720
720                     FORMAT(' PARALLEL GUIDANCE FUNDS IN $M.')
                                   READ(IN,*,ERR=600,END=600) CTC(5)
80    IF ((NC.EQ.1) .OR. (NC.EQ.2)) GO TO 140
                                   PRINT 491
90                                 PRINT 100, NC
100   FORMAT (/,' CONTRACTOR=',I2,' IS INVALID.',/,
     *' PLEASE REENTER CONTRACTOR NUMBER; EITHER "1" OR "2".')
                                   READ(IN,*,ERR=130,END=130) NC
                        GO TO 80
130                                CALL EXITS
                        GO TO 80
C
140   IF ((ICOFG.EQ.1) .OR. (ICOFG.EQ.2)) GOTO 160
                                   PRINT 491
                                   PRINT 145, ICOFG
145   FORMAT (/,' GUIDANCE APPROACH=',I2,' IS INVALID.',/,
     *' PLEASE REENTER GUIDANCE APPROACH; "1" OR "2".')
      READ(IN,*,ERR=150,END=150) ICOFG
      GO TO 140
150   CALL EXITS
      GO TO 140
C
160   IF ((MIS(6).GE.1) .AND. (MIS(6).LE.4)) GOTO 180
      PRINT 491
      PRINT 165, MIS(6)
165   FORMAT (/,' GUIDANCE CONFIGURATION=',I2,' IS INVALID.',/
     *,' RETYPE GUIDANCE CONFIGURATION;"1","2","3",OR "4".')
170   READ(IN,*,ERR=175,END=175) MIS(6)
```

```
      GO TO 160
175   CALL EXITS
      GO TO 160
C
180   XC8 = 1.5
      IF (CTC(8).GE.XC8) GOTO 205
      PRINT 491
      PRINT 185
185   FORMAT (1H0, 'MAINT')
      PRINT 190, CTC(8), XC8
190   FORMAT (1H+, ' ENG FUNDS OF $', F5.2,
     &  'HAVE BEEN RESET TO THE $', F5.2, ' MINIMUM.')
      CTC(8) = XC8
      CALL CHANGE(CTC(8))
      GO TO 180
C
205   XC7 = 0.5
      IF (CTC(7).GE.XC7)  GOTO 230
      PRINT 491
      PRINT 210
210   FORMAT (1H0, 'VALUE')
      PRINT 190, CTC(7), XC7
      CTC(7) = XC7
      PRINT 220, CTC(7)
220   FORMAT(1H+,'VALUE ENG FUNDS = $',F5.2)
      CALL CHANGE(CTC(7))
      GO TO 205
C
230   IF ((MIS(6).EQ.1) .OR. (MIS(6).EQ.3)) GOTO 300
235   IF (CTC(5).LE.0.0) GOTO 360
      PRINT 491
255   PRINT 240
240   FORMAT (1H0, 'PARALLEL DEV FUNDS')
      PRINT 245,CTC(5)
245   FORMAT (1H+, ' OF $', F5.2)
      PRINT 250
250   FORMAT (1H+, 'ARE NOT REQUIRED, AND HAVE BEEN RESET TO $0.0')
      CTC(5) = 0.0
      PRINT 260
260   FORMAT(1X,'DO YOU WISH TO CHANGE GUIDANCE CONFIGURATION: "Y"=YES O
```

```
      *R "N"=NO?')
       READ(IN,54,ERR=255,END=255)ANS
54     FORMAT(A1)
          IF(ANS.EQ.E)CALL EXITS
          IF(ANS.EQ.ANC)GO TO 360
267    PRINT 270
270    FORMAT(1X,'PLEASE RETYPE GUIDANCE CONFIGURATION; "1","2","3" OR "4
      *".')
       CALL CHANGE(MIS(6))
       GOTO 160
C
300    IF (CTC(5).GT.0.0) GOTO 360
       PRINT 491
310    PRINT 240
       PRINT 320
320    FORMAT (1H+, ' ARE REQUIRED.')
       PRINT 335
335    FORMAT(1X,'DO YOU WISH TO CHANGE GUIDANCE CONFIGURATION: "Y"=YES O
      *R "N"=NO?')
       READ(IN,54,ERR=300,END=300)ANS
          IF(ANS.EQ.E)CALL EXITS
          IF(ANS.EQ.ANC)GOTO 280
          CALL CHANGI(MIS(6))
          GOTO 160
280    PRINT 290
290    FORMAT(1X,'INPUT PARALLEL DEVELOPMENT FUNDS; EX: $5,000,000= 5 ')
       READ(IN,*,ERR=300,END=300)CTC(5)
       GO TO 160
C
360    CALL FRTCMS('CLRSCRN ')
       DO 361 INV=1,4
       IF(IPG(INV).EQ.2) GOTO 362
361    CONTINUE
       GOTO 445
610                                CALL EXITS
362    CONTINUE
       PRINT 365,KDP
365    FORMAT(1X,'*****INPUT PAGE 2 DP-',I1,' INFO.*****')
       PRINT 725
725    FORMAT(' MOTOR RELIABILITY.')
```

```
      READ(IN,*,ERR=610,END=610)YC(1)
      PRINT 730
730   FORMAT(' AIRFRAME RELIABILITY.')
      READ(IN,*,ERR=610,END=610)YC(2)
      PRINT 735
735   FORMAT(' LAUNCHER/GSE RELIABILITY.')
      READ(IN,*,ERR=610,END=610)YC(3)
      PRINT 740
740   FORMAT(' FIRE CONTROL ACCURACY.')
      READ(IN,*,ERR=610,END=610)YC(4)
      PRINT 745
745   FORMAT(' GUIDANCE ACCURACY.')
      READ(IN,*,ERR=610,END=610)YC(5)
      PRINT 750
750   FORMAT(' MOTOR QUAL TESTS:20-40.')
      READ(IN,*,ERR=610,END=610)QT(2)
      PRINT 755
755   FORMAT(' AIRFRAME QUAL TESTS:3-9.')
      READ(IN,*,ERR=610,END=610)QT(3)
      PRINT 760
760   FORMAT(' LAUNCHER/GSE QUAL TESTS:2-6.')
      READ(IN,*,ERR=610,END=610)QT(6)
      PRINT 765
765   FORMAT(' FIRE CONTROL QUAL TESTS:2-4.')
      READ(IN,*,ERR=610,END=610)QT(4)
      PRINT 770
770   FORMAT(' GUIDANCE SYSTEM QUAL TESTS:3-9.')
      READ(IN,*,ERR=610,END=610)QT(5)
      PRINT 775
775   FORMAT(' FLIGHT TESTS:10-25.')
      READ(IN,*,ERR=610,END=610)QT(1)
382   IF(QT(2).GE.20..AND.QT(2).LE.40.)GO TO 385
      PRINT 491
      PRINT 383,QT(2)
383   FORMAT(1X,'MOTOR QUALIFICATION TESTS MUST BE GREATER THAN 20 AND L
     *ESS THAN 40. ',/,' YOUR QUAL TESTS= ',F2.0)
              CALL CHANGE(QT(2))
      GO TO 382
385   IF(QT(3).GE.3..AND.QT(3).LE.9.)GO TO 400
      PRINT 491
```

```
      PRINT 390,QT(3)
390   FORMAT(1X,'AIRFRAME QUALIFICATION TESTS MUST BE GREATER THAN 3 AND
     * LESS THAN 9.',/,' YOUR QUAL TESTS= ',F2.0)
               CALL CHANGE(QT(3))
      GO TO 385
400   IF(QT(6).GE.2..AND.QT(6).LE.6.)GO TO 410
      PRINT 491
      PRINT 405,QT(6)
405   FORMAT(1X,'LAUNCHER QUALIFICATION TESTS MUST BE GREATER THAN 2 AND
     * LESS THAN 6.',/,' YOUR QUAL TESTS= ',F2.0)
               CALL CHANGE(QT(6))
      GO TO 400
410   IF(QT(4).GE.2..AND.QT(4).LE.4.)GO TO 420
      PRINT 491
      PRINT 415,QT(4)
415   FORMAT(1X,'FIRE CONTROL QUALIFICATION TESTS MUST BE GREATER THAN 2
     * AND LESS THAN 4.',/,' YOUR QUAL TESTS= ',F2.0)
               CALL CHANGE(QT(4))
      GO TO 410
420   IF(QT(5).GE.3..AND.QT(5).LE.9.)GO TO 430
      PRINT 491
      PRINT 425,QT(5)
425   FORMAT(1X,'GUIDANCE QUALIFICATION TESTS MUST BE GREATER THAN 3 AND
     * LESS THAN 9.',/,' YOUR QUAL TESTS= ',F2.0)
               CALL CHANGE(QT(5))
      GO TO 420
430   IF(QT(1).GE.10..AND.QT(1).LE.25.)GO TO 440
      PRINT 491
      PRINT 435,QT(1)
435   FORMAT(1X,'FLIGHT TESTS MUST BE GREATER THAN 10 AND LESS THAN 25.'
     *,/,' YOUR FLIGHT TESTS= ',F2.0)
               CALL CHANGE(QT(1))
      GO TO 430
440   CONTINUE
445   CALL PRTCMS('CLRSCRN ')
      DO 446 INV=1,4
      IF(IPG(INV).EQ.3) GOTO 447
446   CONTINUE
      GOTO 510
620   CALL EXITS
```

```
             CALL CHANGE(CDMAX)
      PRINT 497
497   FCRMAT(1X,'FOR MINIMUM COST,')
             CALL CHANGE(CDMIN)
             GO TO 480
490   IF(TDMAX.GE.TDMIN)GOTC 500
      PRINT 491
491   FCRMAT(' *****WARNING*****')
      PRINT 496,TDMAX,TDMIN
496   FORMAT(1X,'FLIGHT TEST LATEST WEEK IS LESS THAN EARLIEST WEEK; LAT
     *EST=',F6.0,'; EARLIEST=',F6.0)
             CALL CHANGE(TDMAX)
      PRINT 498
498   FCRMAT(1X,'FOR MINIMUM TIME,')
             CALL CHANGE(TDMIN)
             GO TO 490
500   IF(RSMAX.GE.RSMIN)GOTC 510
      PRINT 491
      PRINT 505,RSMAX,RSMIN
505   FORMAT(1X,'MAXIMUM SYSTEM RELIABILITY IS LESS THAN MINIMUM RELIABI
     *LITY: MAX=',F6.C,'; MIN=',F6.0)
      PRINT 506
506   FORMAT(1X,'FOR MAXIMUM RELIABILITY,')
            CALL CHANGE(RSMAX)
      PRINT 507
507   FORMAT(1X,'FOR MINIMUM RELIABILITY,')
            CALL CHANGE(RSMIN)
            GO TO 500
510   CAIL FRTCMS('CIRSCRN ')
      DO 511 INV=1,4
      IF(IPG(INV).EQ.4) GOTC 512
511   CCNTINUE
      RETURN
630   CAIL EXITS
512   CCNTINUE
      PRINT 520,KDP
520   FORMAT(' *****INFUT PAGE 4 DP-',I1,' INFO.*****'//)
      PRINT 860
860   FORMAT(' MAXIMUM IMPACT ERROR EXPECTED.')
      READ(IN,*,ERR=630,END=630)ESMAX
```

```
447     CONTINUE
        PRINT 450,KDP
450     FORMAT (1X,'*****INPUT PAGE 3 DP-',I1,' INFO.*****'//)
        PRINT 800
800     FORMAT(' MAXIMUM COST IN $M.')
        READ(IN,*,ERR=620,END=620) CDMAX
        PRINT 805
805     FORMAT(' MINIMUM COST IN $M.')
        READ(IN,*,ERR=620,END=620) CDMIN
806     PRINT 810
810     FORMAT(' INCENTIVE FEE % FOR DEVELOPMENT COST.')
        READ(IN,*,ERR=620,END=620) FCDMIN
        IF(FCDMIN.LT.0..OR.FCDMIN.GT.15.) GOTO 806
        PRINT 815
815     FORMAT(' LATEST WEEK FOR FLIGHT TEST COMPLETION.')
        READ(IN,*,ERR=620,END=620) TDMAX
        PRINT 820
820     FORMAT(' EARLIEST WEEK FOR FLIGHT TEST COMPLETION.')
        READ(IN,*,ERR=620,END=620) TDMIN
        PERTOT=15.-FCDMIN
821     PRINT 825
825     FORMAT(' INCENTIVE FEE % FOR FLIGHT TEST COMPLETION')
        READ(IN,*,ERR=620,END=620) FTDMIN
        IF(FTDMIN.LT.0..OR.FTDMIN.GT.PERTOT) GOTO 821
        PRINT 830
830     FORMAT(' MAXIMUM RELIABILITY % EXPECTED.')
        READ(IN,*,ERR=620,END=620) RSMAX
        PRINT 835
835     FORMAT(' MINIMUM RELIABILITY % EXPECTED.')
        READ(IN,*,ERR=620,END=620) RSMIN
        PERTOT=15.-FCDMIN-FTDMIN
836     PRINT 840
840     FORMAT(' INCENTIVE FEE % FOR RELIABILITY.')
        READ(IN,*,ERR=620,END=620) FRSMAX
        IF(FRSMAX.LT.0.CR.FRSMAX.GT.PERTOT) GOTO 836
480     IF(CDMAX.GE.CDMIN) GOTO 490
        PRINT 491
        PRINT 495,CDMAX,CDMIN
495     FORMAT(1X,'MAXIMUM COST IS LESS THAN MINIMUM COST: MAX =',F4.1,
       *';MIN =',F4.1)
```

```
      PRINT 865
865   FORMAT(' MINIMUM IMPACT ERROR EXPECTED.')
      READ(IN,*,ERR=630,END=630) ESMIN
      FEETOT=15.-FCDMIN-FTDMIN-FRSMAX
      PRINT 870
870   FORMAT(' INCENTIVE FEE % FOR IMPACT ACCURACY:')
C     PRINT 871,PERTOT
C 871   FORMAT(' *****REMAINING FEE PERCENTAGE FROM 15% TOTAL IS ',F4.1)
      READ(IN,*,ERR=512,END=512) FESMIN
      PRINT 875
875   FORMAT(' INPUT DESIRED WEEK FOR COMPLETION OF DEPLOYMENT.')
      READ(IN,*,ERR=630,END=630) DD(13)
545   IF(ESMAX.GE.ESMIN)GOTO 553
      PRINT 550,ESMAX,ESMIN
550   FORMAT(1X,'*****WARNING*****MAXIMUM IMPACT ERROR IS LESS THAN MINI
     *MUM IMPACT ERROR:',/,' MAX ERROR =',F6.0,'; MIN ERROR =',F6.0)
      PRINT 552
552   FORMAT(1X,'FOR MINIMUM ERROR, ')
              CALL CHANGE(ESMAX)
      PRINT 551
551   FORMAT(1X,'FOR MINIMUM ERROR, ')
              CALL CHANGE(ESMIN)
              GO TO 545
553   CALL PRCTCK
      RETURN
      END
C *****************************************************************
      SUBROUTINE INPUT2
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &    ,    IFAC(48),    NT(48),     IRC(49),     IDDT(48)
     &    ,    IFD(48),    ITERM(48),    QTY(48),     CMAT(48)
      COMMON   RESS(48),    RESU(48),     RESB(48),     ITYPE(48)
     &    ,    PERC(48),    TDUE(48),     DEC(3),     Y(8)
     &    ,    IPRC(48),    RESC(48),     YF(48),     RESA(48)
     &    ,    TSTAR(48),    CL(6,5),     ALN(48),     CT(48)
     &    ,    TFIN(48),    TMIN(48),     IQ(48),     CUN(30)
      COMMON   RESMIN(30),    RESMAX(30),    AK(30),     ISW(3)
     &    ,    ITAB(25),    NN(8),    N(8),    INDX(8)
     &    ,    RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
```

```
      &    ,     PER(6) ,    PVR (6),    CD,     ICD,    INA,INF,IDP
       COMMCN CPUMAX,CPUMIN, FCPMIN,X,NTEMP(25) ,COV,C10,TRADET,TRADER,
      &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
       CCMMCN    COSTP2,    COSTOT,    ITCT,    ALPHA,    PRICR
      &    ,     TRADEP,    TF,    WA,    IACT(48,2),    IPC(48,5)
      &    ,     TDS(48,2) ,    JMIN,    IC,    IFIG,    KDP,    MR
      &    ,     NC,ICOFG,KP,KKPP,ITEAM
       COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
      &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
       COMMON/STORED/ STAT(5),NDUMP(5,5),ADUMP(5,28),DPCOST,MAXDP
       REAL*8 STAT
       REAL*8 FINAL/'FINAL'/,INTEN/'PROPOSED'/
       INTEGER E/'E'/,C/'O'/,C/'C'/,IN/5/,ANO/'N'/,ANS,YES/'Y'/
       DIMENSION MEMO(48,2)
C         - - - 1972 MAR 23
       JDP=KDP-1
       IF(JDP.LT.3) JDP=3
       NSTTP = 17
       NTF = 5
C         - - - >= DP-4 EDIT'  - - - -
C
C
930    IF(CTC(8).GE.ADUMP(JDP,1))GOTO 940
       CTC(8)=ADUMP(JDP,1)
       PRINT 935,JDP,JDP,ADUMP(JDP,1),KDP
  935 FORMAT(1X,' MAINTENANCE ENG FUNDS ARE LESS THAN THOSE AT DP-',I1,
      &/,1X, 'AND HAVE BEEN RESET TO THE DP-',I1,' FIGURE:$', F5.2,'.',/,
      *' DO YOU WISH TC RESET DP-',I1,' MAINTENANCE ENGINEERING FUNDS:"Y"
      */"N"?')
             READ(IN,2896,ERR=2888,END=2888)ANS
 2896        FORMAT(A1)
             IF(ANS.EQ.E)CALL EXITS
             IF(ANS.EQ.ANO)GO TO 940
             IF(ANS.EQ.YES)GO TC 2889
 2888           CALL EXITS
                GO TO 930
 2889        CALL CHANGE(CTC(8))
             ADUMP(KDP,1)=CTC(8)
             GO TO 930
C
```

```
 940 IF(CTC(7).GE.ADUMP(JDP,2))GOTO 950
     CTC(7)=ADUMP(JDP,2)
     PRINT 1851,KDP,JDP,KDP,CTC(7),KDP
1851 FORMAT(1X,'DP-',I1,' VALUE ENGINEERING FUNDS ARE LESS THAN DP-',I1
    *,'.'/' DP-',I1,' VALUE ENGINEERING FUNDS HAVE BEEN RESET TO: $',F4
    *.2,'.'/' DO YOU WISH TO CHANGE DP-',I1,' VALUE ENGINEERING FUNDS:"
    *Y"/"N"?')
            READ(IN,3896,ERR=3888,END=3888)ANS
3896        FORMAT(A1)
            IF(ANS.EQ.E)CALL EXITS
            IF(ANS.EQ.ANO)GO TO 950
            IF(ANS.EQ.YES)GO TO 3889
3888           CALL EXITS
               GO TO 940
3889        CALL CHANGE(CTC(7))
            GO TO 940
C
950   CALL SETUP(KDP)
      RETURN
      END
C *************************************************************************
      SUBROUTINE INPUT3
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &   ,     IFAC(48),   NT(48),    IRC(49),    IDDT(48)
     &   ,     IFD(48),    ITERM(48),   QTY(48),    CMAT(48)
      COMMON   RESS(48),    RESU(48),    RESB(48),    ITYPE(48)
     &   ,     PERC(48),   TDUE(48),    DEC(3),     Y(8)
     &   ,     IPRC(48),   RESC(48),    YF(48),     RESA(48)
     &   ,     TSTAR(48),   CL(6,5),    ALN(48),    CT(48)
     &   ,     TFIN(48),   TMIN(48),    IQ(48),     CUN(30)
      COMMON   RESMIN(30),   RESMAX(30),   AK(30),    ISW(3)
     &   ,     ITAB(25),   NN(8),    N(8),    INDX(8)
     &   ,     RATIO(8),   AN(9),    ADJ(2,8),    PDR(6)
     &   ,     PER(6),   PVR(6),   CD,   ICD,   INA,INF,IDP
      COMMON CPUMAX,CPUMIN,PCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
     &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
      COMMON   COSTP2,   COSTOT,   ITCT,   ALPHA,   PRICR
     &   ,     TRADEP,   TF,   WA,   IACT(48,2),   IPC(48,5)
     &   ,     TDS(48,2),   JMIN,   IC,   IFIG,   KDP,   MR
```

```
     &    ,      NC,ICOFG,KP,KKPP,ITEAM
      COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
      DIMENSION MEMO(48,2)
      NSTTP = 17
      NTF = 5
 1002 IC=NC
      IF (NTEMP(1)  .EQ.3) GOTO 1919
      IF (IDP .NE. 0) GOTO 1920
C
 1919 NTEMP(2)=3
      IDP=3
      GOTO  (21, 24), NC
   21 GOTO  (22,23), ICOFG
   22 NSTTP = 17
      GOTO 27
   23 NSTTP = 18
      GOTO 27
   24 GOTO  (25,26), ICOFG
   25 NSTTP = 19
      GOTO 27
   26 NSTTP = 20
   27 CONTINUE
      REWIND NSTTP
      DO 20 K=1,13
      READ(NSTTP,100) L,(TAB(K,I,2),I=1,25)
  100 FORMAT(   5X,I2,4X,7F6.2/ 1X,9F6.2/ 1X,9F6.2)
   20 CONTINUE
      READ(NSTTP,102) (TDS(K,1),K=1,13)
  102 FORMAT(   9X,13F4.0)
      READ(NSTTP,103) (ISUBS(I),I=1,48)
  103 FORMAT(   9X,22I2/1X,26I2)
      READ(NSTTP,105) (IFAC(I),I=1,48)
  105 FORMAT(   9X,14I3 /1X,17I3/1X,17I3)
      READ(NSTTP,103) (NT(I),I=1,48)
      READ(NSTTP,108) (YF(I),I=1,48)
  108 FORMAT(   9X,9F5.0 /1X,13F5.0 /1X,13F5.0 /1X,13F5.0)
      READ(NSTTP,103) (IRC(I),I=1,48)
      READ(NSTTP,103) (IDDT(I),I=1,48)
      READ(NSTTP,105) (IQ(I),I=1,48)
```

```
      READ(NSTTP,105)  (IFD(I),I=1,48)
      READ(NSTTP,110)  (ITERM(I),I=1,48)
  110 FORMAT(    9X,48I1)
      READ(NSTTP,108)  (QTY(I),I=1,48)
      READ(NSTTP,111)  (CMAT(I),I=1,48)
  111 FORMAT(    9X,8F6.0 /1X,10F6.0 /1X,10F6.0/
     & 1X,10F6.0/ 1X,10F6.0)
      READ(NSTTP,108)  (RESS(I),I=1,48)
      READ(NSTTP,108)  (RESU(I),I=1,48)
      READ(NSTTP,108)  (RESB(I),I=1,48)
      DO 500 J=1,5
      READ(NSTTP,113)  (IPC(I,J),I=1,48)
  113 FORMAT(    11X,14I3 /  1X,17I3 /1X,17I3)
  500 CONTINUE
      READ(NSTTP,103)  (ITYPE(I),I=1,48)
      READ(NSTTP,115)  (PERC(I),I=1,48)
  115 FORMAT(    7X,8F6.4 /  1X,  10F6.4 /1X,10F6.4 /
     &   1X,10F6.4 /1X,10F6.4)
      READ(NSTTP,108)  (TDUE(I),I=1,48)
      READ(NSTTP,117)  (ALN(I),I=1,48)
  117 FORMAT(    9X,14F3.0 /  1X,17F3.0 /1X,17F3.0)
      READ(NSTTP,105)  (IPRC(I),I=1,48)
      READ(NSTTP,120)  (CUN(I),I=1,29)
  120 FORMAT(    9X,9F5.0 /  1X,10F5.0 /1X,10F5.0)
      READ(NSTTP,120)  (RESMAX(I),I=1,29)
      READ(NSTTP,120)  (RESMIN(I),I=1,29)
      DO 9099 I=1,48
      IF(IFAC(I).GT.30)IFAC(I)=30
 9099 CONTINUE
      CUN(30)  =  .0
      RESMIN(30)  =1.
      RESMAX(30)  =1.
C
C        . . . CALCULATE COL 1 FROM COL 2 OF TABLES . . .
C
      TAB(1,1,1)=300
      TAB(2,1,1)=400
      TAB(3,1,1)=650
      TAB(4,1,1)=114
      TAB(5,1,1)=080
```

```
      TAB(6,1,1)=090
      TAB(7,1,1)=120
      TAB(8,1,1)=300
      TAB(9,1,1)=110
      TAB(10,1,1)=118
      TAB(11,1,1)=128
      TAB(12,1,1)=150
      TAB(13,1,1)=10
      DO 39 I=2,25
      DO 30 K=1,3
   30 TAB(K,I,1)=TAB(K,I-1,1)  + 10
      DO 31 K=5,6
   31 TAB(K,I,1)=TAB(K,I-1,1)  + 1
      TDS(I,2)  =2.
      TAB(12,I,1)=TAB(12,I-1,1)  + 10
      TAB(4,I,1)=TAB(4,I-1,1)+2
      TAB(13,I,1)=TAB(13,I-1,1)+1
      DO 35 K=7,8
   35 TAB(K,I,1)=TAB(K,I-1,1)  +20
      DO 41 K=9,11
   41 TAB(K,I,1)=TAB(K,I-1,1)  +2
   39 CONTINUE
C
      TDS(1,2)  = 2.
      INA=48
      INF =30
C
      DO 3 I=1,20
      DO 4 K=1,25
      TAB(I,K,1)=TAB(I,K,1)*10.0**(TDS(I,1)-5.0)
      TAB(I,K,2)=TAB(I,K,2)*10.0**(TDS(I,2)-5.0)
    4 CONTINUE
      TDS(I,1)=0.0
      TDS(I,2)=0.0
    3 CONTINUE
      IF(NC.NE.2)GOTO 1920
C
C        --- APEX PERFORMANCE BUMP' ---
      H=100.0
      F=0.05
```

```
      TAB(14,25,3)=F
C
      DO 8001 I=1,25
      T1=TAB(1,I,2)
      T2=TAB(2,I,1)
      T3=TAB(3,I,1)
      T4=TAB(4,I,2) *H
      T5=TAB(5,I,2) *H
      T6=TAB(6,I,2) *H
      T9=TAB(9,I,2) *H
      T10=TAB(10,I,2) *H
      T11=TAB(11,I,2) *H
C
      TAB(1,I,2)  =  T1+(F*(1.0-T1))
      TAB(2,I, 1) =T2* (1.0-F)
      TAB(3,I, 1) =T3* (1.0-F)
      TAB(4,I, 2) =(T4- (F*T4) )/H
      TAB(5,I, 2) =(T5- (F*T5) )/H
      TAB(6,I, 2) =(T6- (F*T6) )/H
      TAB(9,I, 2) =(T9- (F*T9) )/H
      TAB(10,I,2)= (T10- (F*T10) )/H
      TAB(11,I,2)= (T11- (F*T11) )/H
C
 8001 CONTINUE
C
 1920 CONTINUE
      RSMAX=RSMAX/100.0
      RSMIN=RSMIN/100.0
      FRSMAX=FRSMAX/100.
      FESMIN=FESMIN/100.
      CDMAX=CDMAX*1000000.
      CDMIN=CDMIN*1000000.
      FCDMIN=FCDMIN/100.
      CPUMAX=CPUMAX*1000.
      CPUMIN=CPUMIN*1000.
      FCPMIN=FCPMIN/100.
      FTDMIN=FTDMIN/100.
C
      DO 13 I=1,5
   13 YC(I)=YC(I)/100.
```

```
      CTC(8)=CTC(8)*1C00000.
      CTC(7)=CTC(7)*1000000.
      CTC(5)=CTC(5)*1C00000.
      IF (MIS(6)-1) 11,12,11
   12 CTC(6)=CTC(5)
      CTC(5)=0.0
   11 CONTINUE
C
      COV=12000.
      IF (NTEMP(2).EQ.4) GO TO 9999
C
      DO 3200I=1,INA
      MEM=IPRC(I)
      IPRC(I)=0
 3051 IF (MEM) 3070,3070,3060
 3060 IF (MEM-48) 3080,3080,3070
 3070 MEM=I
 3080 MEMO(MEM,1)=I
      ALN(I)=ALN(I)/100.0
      TDUE(I)=TDUE(I)/10.0
      CMAT(I)=CMAT(I)*1000.0
 3200 CCNTINUE
C
  800 DC 3290 M=1,INA
      IF (ITYPE(M)-5) 3230,3220,3230
 3220 RESB(M)=RESU(M)/1.5
      GO TO 3231
 3230 RESB(M)=RESU(M)
C
 3231 DO 3280 I=1,5
 3240 IF (IPC(M,I)) 3280,3280,3250
 3250 IF (IPC(M,I)-100) 3270,3270,3260
 3260 IPC(M,I)=IPC(M,I)-100
      GO TO 3240
 3270 KJ=IPC(M,I)
      IPC(M,I)=MEMO(KJ,1)
 3280 CONTINUE
 3290 CCNTINUE
C
      CCV=12000.0
```

```
   80 CONTINUE
 9999 CONTINUE
      RETURN
      END
C ***********************************************************************
      SUBROUTINE ZRW
C
C THIS SUBROUTINE ZEROIZES THE ENTIRE COMMON.
      COMMON IRR(3609)
      DO 4 I=1,3506
    4 IRR(I)=0
      RETURN
      END
C ***********************************************************************
      SUBROUTINE PRESET
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &    ,     IFAC(48),    NT(48),     IRC(49),     IDDT(48)
     &    ,     IFD(48),    ITERM(48),    QTY(48),    CMAT(48)
      COMMON     RESS(48),     RESU(48),     RESB(48),     ITYPE(48)
     &    ,     PERC(48),    TDUE(48),     DEC(3),     Y(8)
     &    ,     IPRC(48),    RESC(48),     YF(48),     RESA(48)
     &    ,     TSTAR(48),     CL(6,5),     ALN(48),     CT(48)
     &    ,     TFIN(48),    TMIN(48),     IQ(48),     CUN(30)
      COMMON     RESMIN(30),     RESMAX(30),     AK(30),     ISW(3)
     &    ,     ITAB(25),    NN(8),    N(8),    INDX(8)
     &    ,     RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
     &    ,     PER(6),    PVR(6),    CD,    ICD,    INA,INF,IDP
      COMMON CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
     &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
      COMMON     COSTP2,     COSTOT,     ITCT,     ALPHA,     PRIOR
     &    ,     TRADEP,    TF,     WA,     IACT(48,2),    IPC(48,5)
     &    ,     TDS(48,2),     JMIN,    IC,    IFIG,    KDP,     MR
     &    ,     NC,ICOFG,KP,KKPP,ITEAM
      COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
      ITAB(25)=  IDP
C        TOTAL FLIGHT TEST FAB. COSTS
      DO 980 M=1,INA
      I=IFD(M)-100
```

```
      IF  (I) 950,950,901
  901 IF  (I-6)  902,902,950
  902 J=IFAC(M)
      CL(I,1)=CL(I,1)+CUN(J)
      CL(I,2)=CL(I,2)+CMAT(M)
      CL(I,3)=CL(I,3)+RESB(M)
      CL(I,5)=ALN(M)
  950 CONTINUE
      IF  (IRC(M)) 980,980,949
  949 IF  (IRC(M)-6)  951,951,980
  951 IF  (ISUBS(M)-8)  952,980,980
  952 IF  (ITYPE(M)-1)  980,953,980
  953 I=IRC(M)
      CL(I,4)=CL(I,4)+RESB(M)
  980 CONTINUE
C
  400 QT(11)=200.0
      QT(12)=140.0
  402 QT(7)=(QT(11)/100.)*5.
      QT(9)=(QT(12)/100.)*5.
C
  404 DO 109 M=1,INA
      I=IQ(M)
      IF  (QTY(M)) 109,109,500
  500 IF  (I-1)  109,103,101
  101 IF  (I-12)  103,103,109
  103 X=QT(I)/QTY(M)
      IF(X)   97, 97,105
  105 IF  (PERC(M)-1.0)  107,106,106
  106 IF  (I-7)  97,107,107
   97 X=1.0
  107 TAB(1,I,3)=X
      QTY(M)=QTY(M)*X
  109 CONTINUE
C
  110 DO 119 M=1,INA
      I=IQ(M)-100
      IF  (I-1)  119,113,111
  111 IF  (I-12)  113,113,119
  113 QTY(M)=QTY(M)*TAB(1,I,3)
```

```
      I=QTY(M)+0.1
      QTY(M)=I
  119 CONTINUE
C
      IF (IDP-4) 120,120,250
  120 DO 129 M=1,INA
      IF (IFD(M)-1)  129,121,122
  121 IF (MIS(6)-4)  124,123,129
  123 QTY(M)=0.0
      GO TO 129
  124 QTY(M)=1.0
      GO TO 129
  122 IF (IFD(M)-2)  129,125,129
  125 IF (MIS(6)-2)  124,123,124
  129 CONTINUE
C          SET END DATES
  250 DO 269 M=1,INA
      IF (PERC(M)-1.0) 257,269,269
  257 TDUE(M)=900.0
      IF (IC(M)-11)  258,260,258
  258 IF (IC(M)-1)  269,252,269
  252 JJ=IFAC(M)
      X=(RESU(M)*(QTY(M)-1.0)**(1.0-ALN(M))/RESMIN(JJ))
      TDUE(M)=DD(1)-X
      IF (TDMIN) 256,256,255
  255 TDUE(M)=TDMIN-X
  256 IF (TDUE(M)) 251,251,269
  251 TDUE(M)=200.0
      GO TO 269
C
  260 DO 268 I=1,10
      A=I
      J=I+3
      JJ=IFAC(M)
      IF (DD(J))  261,261,262
  261 AJJ=RESU(M)*20.0/RESMIN(JJ)
      DD(J)=DD(13)-(10.0-A)*AJJ
  262 TAB(1,I,3) = DD(J) - (RESU(M) * (20.0 * A - 1.0) **
     &    (1.0 - ALN(M)) / RESMIN(JJ))
  268 CONTINUE
```

```
C
  269 CONTINUE
C
  280 I=QT(11)/20.0+.1
  281 I=10
  290 DO 299 J=1,I
      IF (TAB(1,J,3)-TAB(1,1,3)) 291,299,299
  291 TAB(1,1,3)=TAB(1,J,3)
  299 CONTINUE
C
  282 I=3
  310 K=I+1
      DO 319 J=K,10
      IF (TAB(1,J,3)-TAB(1,K,3)) 311,319,319
  311 TAB(1,K,3)=TAB(1,J,3)
  319 CONTINUE
C
  330 DO 339 M=1,INA
      IF (IQ(M)-11) 332,331,332
  331 TDUE(M)=TAB(1,1,3)
  332 IF (IQ(M)-12) 339,333,339
  333 IF (IDP-4) 334,335,335
  334 TAB(1,K,3)=TAB(1,K,3)+(RESU(M)*60.   )/RESMIN(JJ)
      GO TO 337
  335 IF(QT(12)) 339,339,336
  336 TAB(1,K,3)=TAB(1,K,3)+(RESU(M)*QT(11))/RESMIN(JJ)
  337 TDUE(M)=TAB(1,K,3)
  339 CONTINUE
C
      IF(IDP.EQ.4)GOTO 380
  371 ALPHA=1.0
  380 DO 600 M=1,INA
      I=IFD(M)
      IF (I-22) 600,605,604
  604 IF (I-25) 605,605,600
  605 IF (QT(I)) 600,600,610
  610 L=IFAC(M)
      RESU(M)=QT(I)/CUN(L)
  600 CONTINUE
C
```

```
      CCNTINUE
      RETORN
      END
C ***********************************************************************
      SUEROUTINE TRADE
C
      CCMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &    ,    IFAC(48),    NT(48),    IRC(49),    IDDT(48)
     &    ,    IFD(48),   ITERM(48),    QTY(48),    CMAT(48)
      CCMMON   RESS(48),    RESU(48),    RESB(48),    ITYPE(48)
     &    ,    PERC(48),    TDUE(48),    DEC(3),    Y(8)
     &    ,    IPRC(48),    RESC(48),    YF(48),    RESA(48)
     &    ,    TSTAR(48),    CL(6,5),    ALN(48),    CT(48)
     &    ,    TFIN(48),    TMIN(48),    IQ(48),    CUN(30)
      COMMON   RESMIN(30),    RESMAX(30),    AK(30),    ISW(3)
     &    ,    ITAB(25),    NN(8),    N(8),    INDX(8)
     &    ,    RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
     &    ,    PER(6),    PVR(6),    CD,    ICD,    INA,INF,IDP
      CCMMCN CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
     &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
      CCMMON   COSTP2,    COSTO1,    ITCT,    ALPHA,    PRICR
     &    ,    TRADEP,    TF,    WA,    IACT(48,2),    IPC(48,5)
     &    ,    TDS(48,2),    JMIN,    IC,    IFIG,    KDP,    MF
     &    ,    NC,ICOFG,KP,KKPP,ITEAM
      COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
      DO 40 M=1,INA
         IF(IQ(M)-201) 40,20,10
   10    IF(IQ(M)-209)20,20,40
   20    K=IQ(M)-200
         JJ=IFAC(M)
         N(K)=M
         K=NT(M)
         DO 30 I=1,24
            J=I+1
            AAA=ABS(TAB(K,J,2)-TAB(K,I,2))
            TAB(K,I,3)=CUN(JJ)*(TAB(K,J,1)-TAB(K,I,1))/(YF(M)*AAA)
   30    CCNTINUE
   40 CONTINUE
         IF(CDMAX.GT.0)GOTO 100
```

```
        TERM1=0.0
      GO TO 110
  100   TERM1=(CDMAX-CDMIN)/FCDMIN
  110   IF(TDMAX.GT.0)GOTO 120
        TRADET=0.0
      GO TO 130
C ZERNUL WAS ADDED IN MAR 84 TO PREVENT DIVIDE BY ZERO ERRORS.
  120                     ZERNUL=TDMAX-TDMIN
                          IF(ZERNUL.EQ.0.0)ZERNUL=0.00000001
        TRADET=TERM1*FTDMIN/(ABS(ZERNUL))
  130   IF(RSMAX.GT.0)GOTO 140
        TRADER=0.0
      GO TO 150
  140                     ZERNUL=RSMAX-RSMIN
                          IF(ZERNUL.EQ.0.0)ZERNUL=0.00000001
        TRADER=TERM1*FRSMAX/(ABS(ZERNUL))
  150   IF(ESMAX.GT.0)GOTO 160
        TRADEA=0.0
      GO TO 170
  160                     ZERNUL=ESMAX-ESMIN
                          IF(ZERNUL.EQ.0.0)ZERNUL=0.00000001
        TRADEA=TERM1*FESMIN/(ABS(ZERNUL))
  170   IF(CPUMAX.GT.0)GOTO 180
        TRADEP=0.0
      GO TO 190
  180                     ZERNUL=CPUMAX-CPUMIN
                          IF(ZERNUL.EQ.0.0)ZERNUL=0.00000001
        TRADEP=TERM1*FCPMIN/(ABS(ZERNUL))
C
  190 CONTINUE
      RETURN
      END
C ***********************************************************************
      SUBROUTINE DESRES
C
C       73MAR07'  REVISED TO DEGRADE TEST VALUES TO
C       -  EQUIVALENT TABLE VALUES.
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &   ,    IFAC(48),    NT(48),    IRC(49),    IDDT(48)
```

```
&    ,   IFD(48),   ITERM(48),   QTY(48),   CMAT(48)
 COMMON    RESS(48),   RESU(48),   RESB(48),   ITYPE(48)
&    ,   PERC(48),   TDUE(48),   DEC(3),   Y(8)
&    ,   IPRC(48),   RESC(48),   YF(48),   RESA(48)
&    ,   TSTAR(48),   CL(6,5),   ALN(48),   CT(48)
&    ,   TFIN(48),   TMIN(48),   IQ(48),   CUN(30)
 COMMON    RESMIN(30),   RESMAX(30),   AK(30),   ISW(3)
&    ,   ITAB(25),   NN(8),   N(8),   INDX(8)
&    ,   RATIO(8),   AN(9),   ADJ(2,8),   PDR(6)
&    ,   PER(6),   PVR(6),   CD,   ICD,   INA,INF,IDP
 COMMON CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
&TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
 COMMON    COSTP2,   COSTOT,   ITCT,   ALPHA,   PRIOR
&    ,   TRADEP,   TF,   WA,   IACT(48,2),   IPC(48,5)
&    ,   TDS(48,2),   JMIN,   IC,   IFIG,   KDP,   MR
&    ,   NC,ICOFG,KP,KKPP,ITEAM
 COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
&FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
C DIMENSIONED YDC(6) 5/83; TO SATISFY WATFIV COMPILER.
 DIMENSION  YDC(6)
C DELETED KDP=ITAB(25) 2/84; IT WAS RESETTING KDP TO 3 ALL THE TIME.
 M6=MIS(6)
    DO 95 M=1,INA
       IF(TMIN(M).LE.0) TMIN(M)=50.0
       KIQ=IQ(M)
    IF((KIQ.LT.201).OR.(KIQ.GT.208)) GOTO 95
C DESIGN DUBSYSTEMS---------------------------------------------------
       K=KIQ-200
       N(K)=M
    IF((K.NE.5).AND.(K.NE.6))GOTO 11
C GUIDANCE------------------------------------------------------------
       IRC(M)=0
C "A" IS PRIMARY OR ALONE---------------------------------------------
    IF((K.EQ.5).AND.(M6.LE.2))GOTO 10
C "B" IS PRIMARY OR ALONE---------------------------------------------
    IF((K.EQ.6).AND.(M6.GE.3))GOTO 10
    GOTO 11
C--------------------------------------------------------------------
 10    IRC(M)=5
       KP=K
```

```
          JMIN=K
          NT(36)=K
   11   IF(KDP.GT.4) GOTO 95
          XCTC=CTC(K)
        IF(XCTC.LE.0.0) GOTO 95
          L=IFAC(M)
          XREST=XCTC/CUN(L)
          RESU(M)=XREST-RESS(M)
          REST(M)=XREST
          NN(K)=1
       IF(KDP.GT.4) GOTO 95
       IF((K.NE.5).AND.(K.NE.6)) GOTO 95
C BACKUP GUIDANCE------------------------------------------------------------
          IF((K.EQ.5).AND.(M6.EQ.3)) GOTO 408
          IF((K.EQ.6).AND.(M6.EQ.1)) GOTO 408
       GOTC 95
C----------------------------------------------------------------------------
  408     L=NT(M)
          LL=IFAC(M)
          X3=0.3333
        DO 410 J=1,25
  410     TAB(L,J,1)=X3*TAB(L,J,1)
        RESMAX(LL)=X3*RESMAX(LL)
        RESMIN(LL)=X3*RESMIN(LL)
C
   95 CCNTINUE
      IF(JDP.GE.5) GCTO 100
C
C DEGRADE INPUT MIN ACCEPT TO EQUIV TAB VALUES-------------------------------
C         -  FOR INIT SEARCH.  DEGRADE INCENT LIMITERS
C         -  SAME FOR ENDS OF INCENT SEARCHES.
C         -  THROUGH STAT 103.
C
C RELIABILITY FACTORS--------------------------------------------------------
      DQT1 = 1.0 - ((40.0 - QT(2)) * .00162)---------------------------------
      DQT2 = 1.0 - ((9.0 - QT(3)) * .002)
      DQT3 = 1.0 - ((6.0 - QT(6)) * .003)
      DFTR = 1.0 - ((25.0 - QT(1)) * .0065)
C ERRCR FACTORS
      DQT4 = 1.0 + ((4.0 - QT(4)) * .019)------------------------------------
```

```
      DQT5 = 1.0 +  ((9.0 - QT(5)) * .0062)
      DQT6 = DQT5
      DFTE = 1.0 +  ((25.0 - QT(1)) * .005)
C RELIABILITY TEST VAIUES
      YDC(1) = YC(1)  * DQT1  * DFTR-------------------------------------
      YDC(2) = YC(2)  * DQT2  * DFTR
      YDC(3) = YC(3)  * DQT3  * DFTR
      DRSMAX = RSMAX  * DQT1  * DQT2 * DQT3 * DFTR
C ERROR TEST VALUES
      YDC(4) = YC(4)  * DQT4  * DFTE-------------------------------------
      YDC(5) = YC(5)  * DQT5  * DFTE
      YDC(6) = YC(6)  * DQT6  * DFTE
      DESMIN = ESMIN  * DQT4  * DQT5 * DFTE
C-----------------------------------------------------------------------
  103 NR=0
      NE=0
      KA=1
      KZ=3
C       MOTCR, AIRFRAME, GSE/LAUNCHER___________________________________
      ASSIGN 102 TO KGO
   99 DO 101 K=KA,KZ
            M=N(K)
            J=NT(M)
            IF(NN(K).NE.1)  GOTO 9
            IF(K.LE.3) NR = NR+1
            XRESU=RESU(M)
            DO 120 I=1,25
               IF(XRESU.LE.TAB(J,I,1)) GOTO 130
  120       CONTINUE
            I=25
         GOTO 130
C-----------------------------------------------------------------------
    9       XYCK = YDC(K)
            DO 14 I=1,25
               IF((K.LE.3).AND.(XYCK.LE.TAB(J,I,2))) GOTO 130
               IF((K.GT.3).AND.(XYCK.GE.TAB(J,I,2))) GOTO 130
   14       CONTINUE
            I=25
  130    INDX(K)=I
  101 CCNTINUE
```

```
      GOTO KGO, (102,56)
C-----------------------------------------------------------------------
  102 IF(NR.GE.3) GOTC 39
      KA=1
      KZ=3
      ASSIGN 31 TO KGCA
  150 DO 30 K=KA,KZ
         M=N(K)
         J=NT(M)
         I=INDX(K)
         RATIO(K)=TAE(J,I,3)
         Y(K)=TAB(J,I,2)
   30 CONTINUE
      GOTC KGOA, (31,38)
C-----------------------------------------------------------------------
   31 XY1=Y(1)
      XY2=Y(2)
      XY3=Y(3)
      IF ((XY1 * XY2 * XY3) .GE. DRSMAX) GOTO 159
      DEC(1)=RATIO(1)/(XY2*XY3)
      DEC(2)=RATIO(2)/(XY1*XY3)
      DEC(3)=RATIO(3)/(XY1*XY2)
      GOTO 33
C-----------------------------------------------------------------------
  159 KA=1
      KZ=3
      ASSIGN 44 TO KGCB
  160 DO 35 K=KA,KZ
         M=N(K)
         J=NT(M)
   18    TAB(20,K,1)=NT(M)
            IF(IDP.LE.3) GOTO 1450
C TO PREVENT "PERFORMANCE SLIP"-----------------------------------------
         I=TAB(20,K,2)
            IF(INDX(K).GE.I) GOTO 1450
         INDX(K)=I
            GOTO 1470
C-----------------------------------------------------------------------
 1450    I=INDX(K)
         TAB(20,K,2)=I
```

```
 1470     XRESU=TAB(J,I,1)
          RESU(M)=XRESU
          REST(M)=XRESU+RESS(M)
   35 CONTINUE
      GOTO KGOB, (44,80)
C
C FIND SMALLEST COST/PERFORMANCE FACTOR'--------------------------------
   33 I=0
      DO 34 K=1,3
          IF(NN(K).EQ.1) GOTO 34
C MAX PERFORMANCE NOT REACHED___________________________________________
          XDECK=DEC(K)
          IF(I.GT.0) GOTO 22
          I=1
          DECMIN=XDECK
          MK=K
   34 CONTINUE
   22 IF(XDECK.LT.DECMIN) MK=K
      IF(XDECK.LT.DECMIN) DECMIN=XDECK
      IF(DEC(MK).GE.TRADER) GOTO 39
C THE "SMALLEST" IS LESS THAN RELIABILITY TRADE-OFF FACTOR______________
          K=MK
          M=N(K)
          J=NT(M)
          INDX(K)=INDX(K)+1
      IF(INDX(K).GT.25) INDX(K)=25
          I=INDX(K)
          RATIO(K)=TAB(J,I,3)
          Y(K)=TAB(J,I,2)
      IF(I.LT.25) GOTO 31
C MAX PERFORMANCE REACHED
          RATIO(K)=9.99*10**9-------------------------------------------
          NN(K)=1
          NR=NR+1
          IF(NR.LT.3) GOTO 31
C
   39         KA=1
              KB=3
              ASSIGN 44 TO KGOB
              GOTO 160
```

```
C
   44 IF(TRADEA.LE.0) NE=2
C        NO INCENTIVE FOR ACCURACY-------------------------------------------
          KA=4
          KZ=6
C FIRE CONTROL, GUIDANCE A, GUIDANCE B------------------------------------
          ASSIGN 56 TO KGO
          GOTO 99
   56     NE=NE+NN(4)+NN(KP)
      IF(NE.GE.2) GOTO 61
          KA=4
          KZ=6
          ASSIGN 38 TO KGOA
          GOTO 150
C
   38 IF((Y(4)+Y(KP)).GT.(DESMIN /YF(M))) GO TO 62
   61     KA=4
          KZ=6
          ASSIGN 80 TO KGOB
          GOTO 160
C
   62 IF(RATIO(4).LE.RATIO(KP)) GOTO 66
      IF(NN(KP).EQ.1) GOTO 68
   64     MK=KP
          GOTO 70
   66 IF(NN(4).EQ.1) GOTO 64
   68     MK=4
C
   70 IF(RATIO(MK).GE.TRADEA) GOTO 61
C
          K=MK
          M=N(K)
          J=NT(M)
          INDX(K)=INDX(K)+1
      IF(INDX(K).GT.25) INDX(K)=25
          I=INDX(K)
          RATIO(K)=TAB(J,I,3)
          Y(K)=TAB(J,I,2)
      IF(I.LT.25) GOTO 38
C MAX PERFORMANCE REACHED-------------------------------------------------
```

```
         RATIO(K)=9.99*10**9
         NN(K)=1
         NE=NE+1
      IF(NE.LT.2) GOTO 38
         GOTO 61
  80     KKPP=1
         GOTO 9999
 100     KKPP=2
9999  RETURN
      END
C ************************************************************************
      SUBROUTINE DESRET
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &    ,     IFAC(48),    NI(48),     IRC(49),     IDDT(48)
     &    ,     IFD(48),   ITERM(48),     QTY(48),     CMAT(48)
      COMMON     RESS(48),     RESU(48),     RESB(48),     ITYPE(48)
     &    ,     PERC(48),    TDUE(48),     DEC(3),      Y(8)
     &    ,     IPRC(48),    RESC(48),     YF(48),      RESA(48)
     &    ,     TSTAR(48),    CL(6,5),     ALN(48),      CT(48)
     &    ,     TFIN(48),    TMIN(48),     IQ(48),      CUN(30)
      CCMMCN     RESMIN(30),     RESMAX(30),     AK(30),      ISW(3)
     &    ,     ITAB(25),    NN(8),    N(8),     INDX(8)
     &    ,     RATIO(8),    AN(9),    ADJ(2,8),     PDR(6)
     &    ,     PER(6),   PVR(6),    CD,    ICD,    INA,INF,IDP
      COMMCN CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
     &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
      COMMON     COSTP2,    COSTOT,     ITCT,     ALPHA,     PRIOR
     &    ,     TRADEP,    TF,     WA,     IACT(48,2),     IPC(48,5)
     &    ,     TDS(48,2),     JMIN,    IC,     IFIG,     KDP,     MR
     &    ,     NC,ICOFG,KP,KKPP,ITEAM
      COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
  80  K=7
      M=N(K)
      J=NI(M)
      ZRSU1=RESU(M)
      IF(IDP.GE.5) GOTO 90
      IF(ZRSU1.LE.TAB(17,11,3)) GOTO 2643
  88  DO 89 I=1,25
```

```
      IF(ZRSU1.LE.TAB(J,I,1)) GOTO 960
   89 CONTINUE
      I=25
      GOTO 965
C
  960 IF(I.EQ.1) GOTO 962
      ZTB1=TAB(J,I-1,1)
      ZTB2=TAB(J,I,1)
      ZTB3=TAB(J,I-1,2)
      ZTB4=TAB(J,I,2)
      ALPHA=ZTB3-(((ZRSU1-ZTB1)/(ZTB2-ZTB1))*(ZTB3-ZTB4))
      GOTO 970
C
  962 ZRSU1=TAB(J,I,1)
  965 ALPHA=TAB(J,I,2)
  970 TAB(17,11,3)=ZRSU1
      TAB(17,12,3)=ALPHA
      TAB(20,7,1)=J
      TAB(20,7,2)=I
      GOTO 76
C
 2643 RESU(M)=TAB(17,11,3)
      ALPHA=TAB(17,12,3)
C
   76 C10 =.0
      DO 850 K=1,2
      M=N(K)
      J=NT(M)
      I=INDX(K)
      RESU(M)=TAB(J,I,1)
      JJ=IRC(M)
      AL=CL(JJ,5)
      AL=.1**AL
      CTCT = CL(JJ,2) + AL * CL(JJ,1) * CL(JJ,3) *
     & (1.0 + 0.5 * RESU(M) / CL(JJ,4))
      C10=C10+CTOT
  850 CONTINUE
C
      K=KP
      M=N(K)
```

```
      J=NT(M)
      I=INDX(K)
      RESU(M)=TAB(J,I,1)
      JJ=IRC(M)
      AL=CL(JJ,5)
      AL=.1**AL
      CTCT = CL(JJ,2) + AL * CL(JJ,1) * CL(JJ,3) *
     &   (1.0 + 0.5 * RESU(M) / CL(JJ,4))
      C10 = C10 + CTOT
      AL=CL(6,5)
      AL=.1**AL
      C10=C10+CL(6,2)+AL*CL(6,1)*CL(6,3)
      TAE(20,25, 1)=C10
      C10 = C10*ALPHA
C
      K=7
      M=N(K)
C        CAIL IN VALUE ENGINEERING RESOURCES FROM TABLE'
   90 XRESU=TAB(17,11,3)
      RESU(M)=XRESU
      REST(M)=XRESU+RESS(M)
      ALPHA=TAB(17,12,3)
C        CCMPUTE UNIT COST OF PRODUCTION'
      C10=TAB(20,25,1)*ALPHA
      TAE(20,25, 1)=C10
C        FIND AVAILABILITY FACTOR (ZULU) FROM MAINT. ENG'
      K=8
      M=N(K)
      J=NT(M)
      ZRSU2=RESU(M)
      IF(IDP.GE.5) GCTO 86
      IF(ZRSU2.LE.TAE(17,13,3)) GOTO 86
C                 IF CURRENT MAINT. ENG. .LE. PREVIOUS, DC NOT SEARCH
      DO 85 I=1,25
      IF(ZRSU2.LE.TAE(J,I,1)) GCTO 84
   85 CCNTINUE
      I=25
      GCTO 975
C
   84 IF(I.EQ.1) GOTC 972
```

```
C         INTERPOLATE'
      ZTB5=TAB(J,I-1,1)
      ZTB6=TAB(J,I,1)
      ZTB7=TAB(J,I-1,2)
      ZTB8=TAB(J,I,2)
      ZULU=ZTB7-(((ZRSU2-ZTB5)/(ZTB6-ZTB5))*(ZTB7-ZTB8))
      GOTO 980
C         STORE MAINT. ENG., ZULU, + TABLE ID
  972 ZRSU2=TAB(J,I,1)
  975 ZULU=TAB(J,I,2)
  980 TAB(17,13,3)=ZRSU2
      TAB(17,14,3)=ZULU
      TAB(20,8,1)=J
      TAB(20,8,2)=I
      GOTO 985
C
   86 RESU(M)=TAB(17,13,3)
  985 REST(M)=RESU(M)+RESS(M)
C
  100 CONTINUE
      RETURN
      END
C ***********************************************************************
      SUBROUTINE SE
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &   ,     IFAC(48),    NT(48),    IRC(49),    IDDT(48)
     &   ,     IFD(48),   ITERM(48),    QTY(48),    CMAT(48)
      COMMON   RESS(48),    RESU(48),    RESB(48),    ITYPE(48)
     &   ,     PERC(48),    TDUE(48),    DEC(3),    Y(8)
     &   ,     IPRC(48),    RESC(48),    YF(48),    RESA(48)
     &   ,     TSTAR(48),    CL(6,5),    ALN(48),    CT(48)
     &   ,     TFIN(48),    TMIN(48),    IQ(48),    CUN(30)
      COMMON   RESMIN(30),    RESMAX(30),    AK(30),    ISW(3)
     &   ,     ITAB(25),    NN(8),    N(8),    INDX(8)
     &   ,     RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
     &   ,     PER(6),    PVR(6),    CD,    ICD,    INA,INF,IDP
      COMMON CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
     &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
      COMMON   COSTP2,   COSTOT,    ITCT,    ALPHA,    PRICR
```

```
     &   ,     TRADEP,   TF,    WA,    IACT(48,2),    IPC(48,5)
     &   ,     TDS(48,2),    JMIN,    IC,    IFIG,    KDP,    MR
     &   ,     NC,ICOFG,KP,KKPP,ITEAM
      COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
      DATA IN/5/
      DO 290 L=1,INF
      AK(L)=0.5*CUN(L)*RESMIN(I)-COV
      RESC(I)=RESMIN(I)
      IACT(L,1)=0
      IACT(I,2)=0
  290 CONTINUE
C         SET ECP
C         MODULE 4
C         ADD DESIGN RESOURCES
      DO 390 M=1,INA
      IF (ITYPE(M)-1) 390,310,390
  310 I=IRC(M)
      IF (I) 390,390,320
  320 ADJ(1,I)=ADJ(1,I)+RESU(M)
      ADJ(2,I)=ADJ(2,I)+RESB(M)
  390 CONTINUE
C         MODULE 2
C         SETS PRC COUNTERS FOR DUE DATE CALCULATION,ADJUSTS FAB
C         RESOURCES AND SETS PERC COUNTERS
      DO 3901 M=1,INA
 3901 IPRC(M)=0
C
      DO 490 M=1,INA
      DO 440 I=1,5
      IF(IPC(M,I)) 440,440,410
  410 J=IPC(M,I)
      IF (ISUBS(M)-8) 430,420,420
  420 IF (ISUBS(M)-ISUBS(J)) 440,430,440
  430 IPRC(J)=IPRC(J)+1
  440 CONTINUE
C
  450 IF (IRC(M)) 470,470,460
  460 IF (ITYPE(M)-5) 470,465,470
  465 J=IRC(M)
```

```
      RESU(M)=(RESB(M)+(ADJ(1,J)/ADJ(2,J))*0.5*RESB(M))*ALPHA
  470 REST(M)=RESS(M)+RESU(M)*(QTY(M)**(1.0-ALN(M)))
      PERC(M)=0.0
      IF (CTY(M)-0.1) 471,471,475
  471 PERC(M)=1.0
      REST(M)=RESA(M)
      IF (TFIN(M)) 1000,1000,490
 1000 TFIN(M)=CD+1.0
      GO TO 490
C
  475 IF (RESA(M)+0.01-REST(M)) 477,471,471
  477 IF (RESA(M)-RESS(M)-RESU(M)) 490,490,476
  476 TERM1 = (((RESA(M) - RESS(M)) / RESU(M)) **
     &     (1.0 / (1.0 - ALN(M)))) / QTY(M)
      IF (TERM1) 490,490,480
  480 IF (TERM1-1.0) 481,481,471
  481 PERC(M) = TERM1
  490 CONTINUE
C         MODULE 3
C         SETS DUE DATES
  590 KTR101=0
      DO 585 MINA=1,INA
      M=(INA-MINA)+1
      IF (IPRC(M)) 585,510,585
  510 IF (PERC(M)-1.0) 520,585,585
  520 KTR101=KTR101+1
      L=IFAC(M)
      IF(L.GT.30)L=30
      IPRC(M)=-1
      AA=TDUE(M)+(REST(M)-RESS(M)-RESU(M))/RESMIN(L)
      TERM3=TDUE(M)-(RESS(M)+RESU(M))/RESMIN(L)
      TERM4=AA-TDUE(M)
C
      DO 570 I=1,5
      J=IPC(M,I)
      IF (J) 580,580,530
  530 IF (ISUBS(M)-8) 540,535,535
  535 IF (ISUBS(M)-ISUBS(J)) 570,540,570
    1 PRINT 1001,M,J,K,I
 1001 FORMAT(4I3)
```

```
      GO TO 2
C
  540 K=IFAC(J)
      IF(K.GT.30)K=30
      IF (J.GT.48 )GO TO 1
    2 CONTINUE
      TERM1=(REST(J)-RESS(J)-RESU(J))/RESMIN(K)
      IF (TERM1-TERM4) 545,545,550
  545 DDD=TERM3
      GO TO 555
C
  550 DDD=TERM3-(TERM1-TERM4)
  555 IF (TDUE(J)-DDD) 565,565,560
  560 TDUE(J)=DDD
  565 IPRC(J)=IPRC(J)-1
C
  570 CONTINUE
  580 TDUE(M) =AA
C
  585 CONTINUE
      IF(KTR101.GT.0) GOTO 590
C        FIX DATA REDUCTION,FLIGHT TEST
      DO 850 M=1,INA
      IF (TDUE(M)-898.0) 850,810,810
  810 I=IPC(M,1)
      IF (I) 850,850,820
  820 L=IFAC(M)
      TDUE(M)=TDUE(I)+REST(M)/RESMIN(L)
  850 CONTINUE
C        MODULE 5
C        SET ACTIVITY POINTERS BY FACILITY
      DO 690 M=1,INA
      IF (PERC(M)-1.0) 610,690,690
  610 I=IFAC(M)
  611 IF (ISUBS(M)-8) 620,640,640
  620 IF (IACT(I,1)) 641,630,641
  630 AK(I)=AK(I)-TRADET
      GO TO 641
C
  640 AK(I)=-5000.0
```

```
  641 IF (IACT(I,1)) 660,650,660
  650 IACT(I,1)=M
      GO TO 690
C
  660 MM=IACT(I,1)
      IF (TDUE(MM)-TLUE(M))  670,670,680
  670 IACT(I,2)=M
      GO TO 690
C
  680 IACT(I,2)=MM
      IACT(I,1)=M
  690 CCNTINUE
C         SET PROCUREMENT DASHPCTS - TAB(17,1,3)= BLOCK1, TAB(17
      TAE(17,3,3)=0.0
      TAE(17,4,3)=0.0
      DO 700 M=1,INA
      J =3
      IF(ISUBS(M)-8)  700,701,702
  702 J=J+1
  701 L= IFAC(M)
      XX=REST(M)
      TAM=CMAT(M)*QTY(M)*ALPHA
      X=0.0
      IF (CT(M))  715,715,710
  710 X=CT(M)-TAM
      XX=REST(M)-RESA(M)
  715 X=X+TAM+XX*CUN(L)
      TAE(17,J,3)=TAE(17,J,3)+X
  700 CCNTINUE
C
      FETUFN
      END
C ***************************************************************************
      SUEROUTINE MOD9
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &   ,    IFAC(48),    NT(48),    IRC(49),    IDDT(48)
     &   ,    IFD(48),   ITERM(48),    QTY(48),    CMAT(48)
      CCMMON   RESS(48),    RESU(48),    RESB(48),    ITYPE(48)
     &   ,    PERC(48),    TDUE(48),    DEC(3),    Y(8)
```

```
     &   ,      IPRC(48),    RESC(48),    YF(48),    RESA(48)
     &   ,      TSTAR(48),    CL(6,5),    ALN(48),    CT(48)
     &   ,      TFIN(48),    TMIN(48),    IQ(48),    CUN(30)
      COMMON    RESMIN(30),    RESMAX(30),    AK(30),    ISW(3)
     &   ,      ITAB(25),    NN(8),    N(8),    INDX(8)
     &   ,      RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
     &   ,      PER(6),    PVR(6),    CD,    ICD,    INA,INF,IDP
      COMMON CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
     &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
      COMMON    COSTP2,    COSTOT,    ITCT,    ALPHA,    PRIOR
     &   ,      TRADEP,    TF,    WA,    IACT(48,2),    IPC(48,5)
     &   ,      TDS(48,2),    JMIN,    IC,    IFIG,    KDP,    MR
     &   ,      NC,ICOFG,KP,KKPP,ITEAM
      COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TIMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
      JJJ=0
   67 JJJ=JJJ+1
      IF (JJJ-250) 65,60,60
C       . . . 60 IS THE RETURN . . .
   65 CONTINUE
      DO 61 I=1,3
      ISW(I)=0
   61 CONTINUE
      CD=CD+1.0
      ICD=ICD+1
C
C         . . . FROM HERE TO 22 IS EXTRACTED FROM
C          MOD7   IN DMS PROGRAM . . . . . .
C
      DO 64 M=1,INA
      L = IFAC(M)
      MIACT = IACT(L,1)
      IF (MIACT.LE.0) GOTO 64
      IF (MIACT.NE.M) GOTO 64
    4 IF (TSTAR(M)-CD)5,5,8
    5 PRIOR=1.0
      X=0.0
C
      DO 7 I=1,5
      J=IPC(M,I)
```

```
      IF (J.LE.0) GOTO 7
      KK = 0
      IF (I.LT.5) KK= IPC(M,I+1)
    6 IF (ITERM(J)-IDP-1) 10,8,10
   10 IF (PERC(J)-PRIOR) 11, 12, 12
   11 PRIOR=PERC(J)
   12 IF(TSTAR(J)-X) 13,13,14
   14 X=TSTAR(J)
   13 IF (KK) 15,7,15
   15 IF(PERC(J)-1.0) 16,7,16
   16 IF (PERC(M)) 7,7,17
   17 JJ=IFAC(J)
      XX=((PERC(J)-PERC(M))*REST(J))/RESC(JJ)
      IF(XX- TMIN(J)) 18,7,7
   18 TMIN(J)=XX
    7 CONTINUE
C
      IF(X-TSTAR(M)) 19,19,20
   20 TSTAR(M)=X
   19 IF(PERC(M)-PRIOR) 21,8 ,8
   21 WA=RESMIN(L)
      IF (AK(L)) 22,27,27
   22 CONTINUE
C
C        . . . FROM HERE TO 26 LESS ONE IS EXTRACTED
C        FROM  MOD11  IN DMS PROGRAM . . . . . .
C
      T=((REST(M)-RESA(M))/RESMIN(L))+CD
      IF(TDUE(M) .LE.CD) GO TO 83
      IF(TDUE(M) .GE.T) GO TO 82
      TERM1=(REST(M)-RESA(M))/(TDUE(M)-CD)
C        . . . CM (A TYPO) CHANGED TO CD 1970 MARCH 25 . . .
      IF(RESMAX(L) .GT.TERM1) GO TO 84
   83 RESC(I) =RESMAX(L)
      GO TO 82
   84 RESC(L) = TERM1
   82 CONTINUE
      IF( RESC(L) .GE. RESMIN(I)) GO TO 26
      RESC(I) = RESMIN(L)
C
```

```
C          . . . FROM HERE TO 100 IS EXTRACTED FROM MOD7
C          IN DMS PROGRAM . . . . . . . . . . . . . .
C
   26 WA=RESC(L)
   27 X=0.0
      I=QTY(M)+0.1
      IF (I-1) 31,50,23
   31 X=1.0
      WA=0.0
      REST(M)=0.0
      GO TO 24
C
   50 IF (PRIOR-1.0) 8,23,23
   23 TERM1=REST(M)-RESA(M)
      IF (TERM1-.01-RESC(L)) 25,25,32
   25 RESA(M)=REST(M)
      WA=TERM1
      PERC(M)=1.0
      GO TO 100
C
   32 IF (RESA(M)-RESS(M)-RESU(M)) 30,30,40
   40 X =((((RESA(M)-RESS(M)+WA)/RESU(M))**(1.0/(1.0-ALN(M))
     &   ))/
     &    (QTY(M)))
   30 IF (X-PRIOR) 24,8,8
   24 RESA(M)=RESA(M)+WA
      PERC(M)=X
  100 CONTINUE
C
C          . . . FROM HERE TO 93 PLUS TWO IS EXTRACTED FROM
C          MOD8 IN DMS PROGRAM . . . . . . . . . . .
C
      I=0
      IF(ISUBS(M)-8) 75,76,77
   77 I=I+1
   76 I=I+1
   75 I=I+1
      IF (RESA(M) .NE.WA) GO TO 71
      CTEST(I) =CTEST(I) +(CMAT(M)*QTY(M))*ALPHA
      CT(M)=CT(M)+(CMAT(M)*QTY(M))*ALPHA
```

```
      TSTAR(M)=CD
C
   71 ISW(I) =ISW(I)+1
      X=.0
      IJ=IFAC(M)
      IF(WA .LE. RESMIN(IJ)) GO TO 81
      X= WA-RESMIN(IJ)
C
   81 XX=(WA*CUN(IJ)+.5*X*CUN(IJ))
      CT(M) =CT(M)  + XX
      CTEST(I)  =CTEST(I)+XX
      IF(RESA(M)+.1 .IT. REST(M)) GO TO 64
      PERC(M) =1.
      RESA(M)=REST(M)
      TFIN(M)=CD
C
      IA1=IACT(L,1)
      IA2=IACT(L,2)
      IF(M.EQ.IA1)GOTC 101
      IF(M.EQ.IA2)GOTC 102
      PRINT 1002
 1002 FORMAT(' 8**MOD* IACT ERROR ***')
      STCP
  101 IACT(L,1)=IA2
  102 IACT(L,2)=0
C
   93 J=ITERM(M)
      IF(J-IDP .NE. 1) GO TO 64
      ITCT =ITCT-1
C
C         . . . FROM HERE TO 64 IS EXTRACTED FROM MOD7
C         IN DMS PROGRAM . . . . . . . . . . . . . . .
C
    8 CONTINUE
   64 RESC(I)=RESMIN(I)
C         . . . 64 CONTINUE CHANGED TO ABOVE  1970 MARCH 25 . . .
C
C         . . . FROM HERE TO END IS EXTRACTED FROM MOD9
C         IN DMS PROGRAM . . . . . . . . . . . . . . .
      IF (ISW(1) .EQ.0) GO TO 66
```

```
      CTCH(1) = CTOH(1) +COV
   66 CONTINUE
      IF (ITCT .GT.0) GO TO 67
      IF (IDP .NE. 7) GO TO 60
      DC 52 L=1,INF
      IF (IACT(L,1).GT.0)GO TO 67
   52 CCNTINUE
   60 CONTINUE
      FETURN
      END
C *******************************************************************
      SUEROUTINE REPCRT
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &    ,    IFAC(48),    NT(48),    IRC(49),    IDDT(48)
     &    ,    IFD(48),   ITERM(48),    QTY(48),    CMAT(48)
      COMMON    RESS(48),    RESU(48),    RESB(48),    ITYPE(48)
     &    ,    PERC(48),    TDUE(48),    DEC(3),    Y(8)
     &    ,    IPRC(48),    RESC(48),    YF(48),    RESA(48)
     &    ,    TSTAR(48),    CL(6,5),    ALN(48),    CT(48)
     &    ,    TFIN(48),    TMIN(48),    IQ(48),    CUN(30)
      CCMMCN    RESMIN(30),    RESMAX(30),    AK(30),    ISW(3)
     &    ,    ITAB(25),    NN(8),    N(8),    INDX(8)
     &    ,    RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
     &    ,    PER(6),    PVR(6),    CD,    ICD,    INA,INF,IDP
      COMMCN CPUMAX,CFUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
     &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
      COMMON    COSTP2,    COSTO1,    ITCT,    ALPHA,    PRICR
     &    ,    TRADEP,    TF,    WA,    IACT(48,2),    IPC(48,5)
     &    ,    TDS(48,2),    JMIN,    IC,    IFIG,    KDP,    MF
     &    ,    NC,ICOFG,KP,KKPP,ITEAM
      COMMCN RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
      X1=0.
      X2=0.
      X3=0.
      ITAB(24)=ICD
      TAB(17,25,3)=CTEST(1) +CTOH(1)
      DO 11 I=1,3
      IF (CTEST(I)) 12,12,11
```

```
   12 CTEST(I)=1.0
   11 CCNTINUE
C
      X1=(CTEST(1)+CTCH(1))/CTEST(1)
      X2=(CTEST(2)+CTCH(2))/CTEST(2)
      X3=(CTEST(3)+CTCH(3))/CTEST(3)
      DO 1 M=1,INA
      IF(ISUBS(M)-8) 2,1,1
    2 X=X1
      GO TO 5
    5 IF (QTY(M)-1.0) 6,6,9
    9 TDS(M,1)=PERC(M)
      GO TO 10
   20 IDS(M,1) =1.
      GO TO 10
C
    6 IF(REST(M) .EQ. .0) GO TO 20
      TDS(M,1) =RESA(M)/REST(M)
   10 TDS(M,2)=CT(M)*X
    1 CCNTINUE
C
      IDE =7
      ITAB(25) =3
      CCNTINUE
      FETURN
      END
C ******************************************************************
      SUEROUTINE PERF
C
      CCMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &   ,    IFAC(48),    NT(48),    IRC(49),    IDDT(48)
     &   ,    IFD(48),    ITERM(48),    QTY(48),    CMAT(48)
      CCMMON   RESS(48),    RESU(48),    RESB(48),    IIYPE(48)
     &   ,    PERC(48),    TDUE(48),    DEC(3),    Y(8)
     &   ,    IPRC(48),    RESC(48),    YF(48),    RESA(48)
     &   ,    TSTAR(48),    CL(6,5),    ALN(48),    CT(48)
     &   ,    TFIN(48),    TMIN(48),    IQ(48),    CUN(30)
      CCMMON   RESMIN(30),    RESMAX(30),    AK(30),    ISW(3)
     &   ,    ITAB(25),    NN(8),    N(8),    INDX(8)
     &   ,    RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
```

```
      &   ,    PER(6) ,    PVR(6),    CD,    ICD,    INA,INF,IDP
       COMMON CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
      &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
       COMMON   COSTP2,   COSTOT,   ITCT,   ALPHA,   PRICR
      &   ,     TRADEP,   TF,   WA,   IACT(48,2),   IPC(48,5)
      &   ;     TDS(48,2) ,   JMIN,   IC,   IFIG,   KDP,   MR
      &   ,     NC,ICOFG,KP,KKPP,ITEAM
       COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
      &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
       INTEGER IN/5/
       DATA PMA/0./,PAMAX/0./,PAMIN/0./,PDA/0./,PVA/0./,PEA/0./
       PMA=0.
       PAMAX=0.
       PAMIN=0.
       PDA=0.
       PVA=0.
       PEA=0.
       KP=JMIN
   100 DO 118 M=1,INA
       TDS(M,1) =1.
       IF (IC(M)-6) 116,116,118
   116 IF(IQ(M)-1) 118,400,117
   117 I=ISUBS(M)
       AN(I)=1.0+QTY(M)*TDS(M,1)
       AK(I)=1.0+QTY(M)
       GO TO 118
   400 FTEST=QTY(M)*TDS(M,1)
       TTEST=QTY(M)
   118 CONTINUE
C
   119 DO 130 K=1,6
       M=N(K)
       I=ISUBS(M)
       L=TAB(20,K,1)
       J=TAB(20,K,2)
       PRIOR=100.0*TAB(L,J,2)*(1.0-TAB(L,J,2))-1.0
       IF (K-4) 106,106,107
   106 AN(I)=AN(I)+PRICR+FTEST
       AK(I)=AK(I)+PRICR+TTEST
       GO TO 130
```

```
C
  107 IF (K-KP)  110,106,110
  110 AN(6)=AN(I)+PRICR+FTEST
      AK(6)=AK(I)+PRICR+TTEST
      SS=AK(6)-PRIOR-1.0
      PEA=TAB(L,J,2)
      PVA=PEA*(1.0-PEA)/(AN(6)+1.0)
      PDA=SQRT(PVA)
      TEM=EXP(0.1*SS)
      FF=0.915+0.314/TEM
C
      IF (FF-1.1) 510,510,500
  500 FF=1.0
  510 PEA=PEA*FF
  520 PAMIN=PEA-PDA
      IF(PAMIN-0.1) 128,129,129
  128 PAMIN=0.1
  129 PAMAX=PEA+PDA
C
      IF(PAMAX-0.99) 109,109,430
  430 PAMAX=0.99
      PMA= (PEA*AN(6)-1.0)/(AN(6)-2.0)
      IF(PMA-PAMAX) 109,109,515
  515 PMA=PEA+0.33*(PAMAX-PEA)
C
  109 IF(PMA-PAMIN)68,70,70
   68 PMA=PEA-0.33*(PEA-PAMIN)
   70 SPER=PMA
      SPEA=PAMIN
      SPVR=PAMAX
C
  130 CONTINUE
C
      DO 200 I=1,5
C         . . . 200 IS CLOSE TO THE END . . .
      GO TO (171,173,175,177,178),I
C         . . . MOTOR . . .
  171 A=1.067
      B=-1.34
      C=0.1
```

```
      K=1
      J1 = 1
      J2 = 2
      J3 = 3
      GO TO 180
C        . . . AIRFRAME . . .
  173 A=1.029
      B=-0.105
      C=0.1
      K=2
      J1 = 4
      J2 = 5
      J3 = 6
      GO TO 180
C        . . . LAUNCHER / GROUND SUPPORT EQUIPMENT . . .
  175 A=1.029
      B=-0.098
      C=0.1
      K=3
      J1 = 7
      J2 = 8
      J3 = 9
      GO TO 180
C        . . . FIRE CONTROL . . .
  177 A=0.908
      B=0.306
      C=0.1
      K=4
      J1 = 11
      J2 = 10
      J3 = 12
      GO TO 180
C        . . . GUIDANCE . . .
  178 A=0.915
      B=0.314
      C=0.1
      K=KP
C        . . . K=5 CHANGED TO K=KP  1970 MARCH 27 . . .
      J1 = 14
      J2 = 13
```

```
      J3 = 15
C
C        THROUGH STATEMENT 191 REVISED 720123 TO CONFORM
C        WITH DMS PERFOR CHANGE OF 720530.
C        CAUSES FLIGHT TEST RESULTS TO ALWAYS BE COMPUTED
C        ON THE EXPECTED NUMBER OF FLIGHT TESTS.
180   L=TAB(20,K,1)
      LL=TAB(20,K,2)
      XPER = TAB(L,LL,2)
      TABLL = XPER
      PRIOR = 100.0 * TABLL * (1.0 - TABLL) - 1.0
      ANIII = AN(I)
C
C        BLOCK II MOST LIKELY
      SS = AK(I) - PRIOR - 1.0
      TEM = EXP(C * SS)
      FF = (A + B / TEM)
      IF (K .GE. 4) GOTO 535
      IF (FF .LT. 0.90) FF = 0.90
      GOTO 550
  535 IF ((FF * XPER) .GT. 0.99) GOTO 91
  550 XPER = XPER * FF
      IF (XPER .LE. 0.99) GOTO 195
   91 XPER = 0.99
  195 TAB(14,J1,2) = XPER
C
C        BLOCK II DEVIATION
      PV = XPER * (1.0 - XPER) / (ANIII + 1.0)
      PD = SQRT(PV)
      TAB(14,J3,2) = PD
C
C        BLOCK II EXPECTED
      PM = (XPER * ANIII - 1.0) / (ANIII - 2.0)
      PMAX = XPER + PD
      IF (PMAX .GT. 0.99) PMAX = 0.99
      IF (PM .LE. PMAX) GOTO 197
      PM = XPER + 0.33 * (PMAX - XPER)
  197 PMIN = XPER - PD
      IF (PMIN .LT. 0.1) PMIN = 0.1
      IF (PM .GE. PMIN) GOTO 191
```

```
      PM = XPER - 0.33 * (XPER - PMIN)
  191 TAB(14,J2,2) = PM
C        . . . ABOVE 10 STATEMENTS INSERTED  1970 APRIL 1 . . .
C
  200 CONTINUE
C
      TAB(14,16,2)  =TAB(14, 1,2)*TAB(14,4,2)*TAB(14,7,2)
      TAB(14,17,2)  = TAB(14,2,2)*TAB(14,5,2)*TAB(14,8,2)
      TAB(14,19,2) =TAB(14,10,2)+TAB(14,13,2)
      TAB(14,20,2) =TAB(14,11,2)+TAB(14,14,2)
C        . . . ABOVE STATEMENT INSERTED  1970 APRIL 1 . . .
      CONTINUE
      RETURN
      END
C ***********************************************************************
      SUBROUTINE PROSUM
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &    ,      IFAC(48),    NT(48),    IRC(49),    IDDT(48)
     &    ,      IFD(48),   ITERM(48),    QTY(48),    CMAT(48)
      COMMON     RESS(48),    RESU(48),    RESB(48),    ITYPE(48)
     &    ,      PERC(48),    TDUE(48),    DEC(3),    Y(8)
     &    ,      IPRC(48),    RESC(48),    YF(48),    RESA(48)
     &    ,      TSTAR(48),    CL(6,5),    ALN(48),    CT(48)
     &    ,      TFIN(48),    TMIN(48),    IQ(48),    CUN(30)
      COMMON     RESMIN(30),    RESMAX(30),    AK(30),    ISW(3)
     &    ,      ITAB(25),    NN(8),    N(8),    INDX(8)
     &    ,      RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
     &    ,      PER(6),   PVR(6),   CD,   ICD,   INA,INF,IDP
      COMMON CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
     &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
      COMMON     COSTP2,    COSTOT,    ITCT,    ALPHA,    PRICR
     &    ,      TRADEP,    TF,    WA,    IACT(48,2),    IPC(48,5)
     &    ,      TDS(48,2),    JMIN,    IC,    IFIG,    KDP,    MR
     &    ,      NC,ICOFG,KP,KKPP,ITEAM
      COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
      COMMON/STORED/ STAT(5),NDUMP(5,5),ADUMP(5,28),DPCCST,MAXDP
      REAL YFEE(8),FRATIO(8),FYC(5)
      REAL*8 STAT
```

```
      REAL*8 FINAL/'FINAL'/,INTEN/'PROPOSED'/
      INTEGER ANS,AMZ/'A'/,ANO/'N'/,YES/'Y'/,E/'E'/,IN/5/
      TMTABA=0.
      TMTABE=0.
      CTESTA=CTEST(1)-(CT(19)+CT(20))
      FCTOH=CTESTA/CTEST(1)
      CTOH(1)=CTOH(1)*FCTOH
      Y(1)=CTEST(1)+CTOH(1)
      DO 5 J=1,10
      ITAB(J)=TAB(19,J,1)
    5 CCNTINUE
      DO 10 M=1,INA
      IF(IQ(M)-1)10,15,10
   10 CONTINUE
C
      M=INA
   15 Y(2)=TFIN(M)
   25 Y(3)=TAB(14,16,2)
      Y(4)  = TAB(14,20,2)
C         . . . ROW 19 CHANGED TO ROW 20   1970 APRIL 1 . . .
   30 K=4
      M=N(K)
      Y(4)=Y(4)*YF(M)
      Y(5)=C10
C
      PER(1)=CDMAX
      AN(1)=CDMIN
      FYC(1)=FCDMIN
      PER(2)=TDMAX
      AN(2)=TDMIN
      FYC(2)=FTDMIN
      AN(3)=RSMAX
      PER(3)=RSMIN
      FYC(3)=FRSMAX
      PER(4)=ESMAX
      AN(4)=ESMIN
      FYC(4)=FESMIN
      PER(5)=CPUMAX
      AN(5)=CPUMIN
      FYC(5)=FCPMIN
```

```
      SPVA=0.0
      FRATIO(8)=0.0
      Y(1)=Y(1)/1000000.0
      Y(3)=Y(3)*100.0
C
C THIS SET OF STEPS TC LABLE 50 CHECKS THE MOST DESIREAELE VAIUES,
C AN(), VS ACHIEVED VALUES, Y(), AND LEAST DESIREABLE VALUE, PER()
C VS ACHIEVED VALUE IN ORDER TO PLACE THE FEE WITHIN THE LINEAR RATIO
C BOUNDS OF THE DESIRED FEE.
            DO 50 I=1,5
              IF(I.EQ.3)GOTO 48
      IF(Y(I)-AN(I).GE.0..AND.PER(I)-Y(I).GE.0.AND.PER(I).NE.AN(I))
     *FRATIC(I)=FYC(I)*(ABS((PER(I)-Y(I))/(PER(I)-AN(I))))
              IF(Y(I)-PER(I).GT.0.)FRATIO(I)=0.0
              IF(AN(I)-Y(I).GE.0.)FRATIO(I)=FYC(I)
                   GOTO 49
48    IF(Y(3)-PER(3).GE.0..AND.AN(3).NE.PER(3))
     *FRATIC(3)=FYC(3)*(ABS((Y(3)-PER(3))/(AN(3)-PER(3))))
              IF(PER(3)-Y(3).GT.0.)FRATIO(3)=0.0
              IF(Y(3)-AN(3).GE.0.)FRATIO(3)=FYC(3)
49            YFEE(I)=FRATIO(I)*(CDMAX+CDMIN)/200.0
              SPVA=SPVA+YFEE(I)
              FRATIC(8)=FRATIO(8)+FRATIO(I)
50          CONTINUE
C
      TOC=Y(1)+SPVA+DECOST/1000000.0
C
      CAIL FRTCMS('CIESCRN ')
                            IF(MR.NE.0)PRINT 14,ITEAM,MAXDP,STAT(MAXDP)
14    FORMAT(12X,'*********TEAM ',I2,' IS IN DP-',I1,1X,A8,'.*********')
      PRINT 1000,NTEMP(1),ITEAM
 1000 FORMAT('1DP-',I1,' *** DEVELOPMENT CONTRACT ',
     &   'SUMMARY *** TEAM',I3)
      PRINT 1002
 1002 FORMAT(1H-,18X,'INCENTIVE PROVISIONS',11X,
     &   'INCENTIVE ACHIEVEMENTS'/
     &   2X,'INCENTIVE',6X,'WORST',5X,'BEST',4X,'MAX FEE',5X,'ACHIEVED',
     &   4X,'FEE',5X,'FEE %' )
      PRINT 1010
 1010 FORMAT(4X,'AREA',9X,'VAIUE',5X,'VALUE',3X,'ALLCWED',6X
```

```
     &   ,'VALUE',5X,'EARNED',3X,'EARNED',/)
C
      PRINT 1020, CDMAX, CDMIN, FCDMIN, Y(1), YFEE(1),FRATIO(1)
 1020 FORMAT( ' DEV. COST',6X,'$',F5.1,'M  $',F5.1,'M',
     &   F7.1,'%',5X,'$',F6.2,'M    $',F6.3,'M',F7.2,'%')
C
      ITA = TDMAX
      ITI = TDMIN
      IY = Y(2)
      PRINT 1030, ITA, ITI, FTDMIN, IY,YFEE(2),FRATIC(2)
 1030 FORMAT(1H ,'FLT TST COMPL',I6,' WK',I6,' WK',F7.1,
     &   '%',I10, ' WK',3X,'$',F6.3,'M',F7.2,'%')
C
      Y(3) = TAB(14,16,2) * 100.
      PRINT 1040, RSMIN, RSMAX, FRSMAX, Y(3), YFEE(3),FRATIO(3)
 1040 FORMAT(' RELIABILITY',F10.1,'%',F8.1,'%',F7.1,'%',
     &   F12.2,'%',3X,'$',F6.3,'M',F7.2,'%' )
C
      ITA = ESMAX
      ITI = ESMIN
      IY=TAB(14,20,2)*100.
C       . . . IY=Y(4) CHANGED TO ABOVE  1970 APRIL 1 . . .
      CPUMAX = CPUMAX / 1000.
      CPUMIN = CPUMIN / 1000.
      PRINT 1050, ITA, ITI, FESMIN, IY, YFEE(4),FRATIO(4)
 1050 FORMAT(' ACCURACY',7X,I4,'YDS',I6,'YDS',F7.1,'%',
     &   I10,'YDS',3X,'$',F6.3,'M',F7.2,'%')
C
      FEETOT=FCDMIN+FTDMIN+FRSMAX+FESMIN+FCPMIN
      PRINT 1070,FEETOT,SPVA,FRATIO(8)
 1070 FORMAT(1H0,'TOTALS',
     &   28X,
     &   F4.1,'%',
     &   15X,
     &   '$',F7.3,'M',F7.2,'%')
      IF(ABS(FEETOT-15.0).LT..0001)GOTO 1090
      PRINT 1080
 1080 FORMAT(1H ,'TOT MAX FEE ALLOWED DOES NOT EQUAL
     &   15.0% ****')
C
```

```
1090   DPCST=DPCOST/1000000.0
       CONTINUE
       IF(DPCOST.GT.0.00)PRINT 2001,DPCST
2001   FORMAT(1H0,'ADDED COST FOR CONTRACT ANALYSES = $',F5.1,'M')
       PRINT 2000,TOC
 2000  FORMAT(1H0,'TOTAL CONTRACT PRICE             = $',F5.1,'M',/)
                                PAUSE ';PRESS <<ENTER>> TO CONTINUE.'
2025                                       RETURN
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C
C         REPORT TEST COMPONENT COSTS TO THE MONITOR.
C
       ENTRY PROSM2
       CALL GET(KDP)
       ITA = TDMAX
       CALL FRTCMS('CLRSCRN ')
  111  FORMAT (1X,25('-'))
  210  FORMAT ('1COSTS OF TESTS PER UNIT (IN THOUSANDS OF DOLLARS)')
       PRINT 210
  220  FORMAT (1H0, 3X, 'QUALIFICATION TESTS')
       PRINT 220
       Z3 = 1000.0
C
C         MOTOR
       ZYX = ((CT(29) +CT(34)) / QT(2)) / Z3
       PRINT 261, ZYX
  261  FORMAT (1H+, 5X, '        MOTOR :',T23,F8.3)
C
C         AIRFRAME
       ZYX = ((CT(30) +CT(35)) / QT(3)) / Z3
       PRINT 262, ZYX
  262  FORMAT (1H+, 5X, '   AIR FRAME :',T23,F8.3)
C
C         GUIDANCE
       ZYX = ((CT(31) +CT(36)) / QT(5)) / Z3
       PRINT 263, ZYX
  263  FORMAT (1H+, 6X, '   GUIDANCE :',T23,F8.3)
C
C         FIRE CONTROL
       ZYX = ((CT(32) +CT(37)) / QT(4)) / Z3
```

```
      PRINT 264,ZYX
  264 FORMAT (1H+, 5X, 'FIRE CONTROL :',T23,F8.3)
C
C         LAUNCHER
      ZYX = ((CT(33) +CT(38)) / QT(6)) / Z3
      PRINT 265,ZYX
  265 FORMAT (1H+, 6X, '   LAUNCHER :',T23,F8.3)
C
C         FLIGHT TESTS
      ZYX = ((CT(40) +CT(41) +CT(42) +CT(46) +CT(47))
     &   / CT(1)) / Z3
      PRINT 270,ZYX
  270 FORMAT (1H+, 3X, '  FLIGHT TESTS : ',T23,F8.3)
      PRINT 111
C
C         DESIGN FACTORS
C
      PRINT 310
  310 FORMAT (1H0, 'DESIGN FACTORS ACHIEVED')
      PRINT 320
  320 FORMAT (8X,'COMPONENT',T22,'TABLE',T29,'ROW',T35,'FACTOR',T44,
     &'ACHIEVED')
C
C         MOTOR
      KZTB = TAB(20,1,1)
      KZRW = TAB(20,1,2)
      PRINT 330, KZTB, KZRW, TAB(KZTB,KZRW,2),TAB(14,2,2)
  330 FORMAT(12X,'MOTOR :',T23,I2,T29,I2,T34,F8.3,T44,F7.3)
C
C         AIRFRAME
      KZTB = TAB(20,2,1)
      KZRW = TAB(20,2,2)
      PRINT 331, KZTB, KZRW, TAB(KZTB,KZRW,2),TAB(14,5,2)
  331 FORMAT(8X,'AIR FRAME :',T23,I2,T29,I2,T34,F8.3,T44,F7.3)
C
C         GUIDANCE A
      KZTB = TAB(20,5,1)
      KZRW = TAB(20,5,2)
C TMTABA AND TMTABB WERE FORMED TO ALLOW IF STATEMENT TO GENERATE THE
C FOURTH POSITION WRITE STATEMENT VARIABLES.
```

```
      IF (MIS(6) .LE. 2)TMTABA=TAB(14,14,2)
      PRINT 402, KZTB, KZRW, TAB(KZTB,KZRW,2),TMTABA
  402 FORMAT(7X,'GUIDANCE A :',T23,I2,T29,I2,T34,F8.3,T44,F7.3)
C
C         GUIDANCE B
      KZTB = TAB(20,6,1)
      KZRW = TAB(20,6,2)
      IF (MIS(6) .GE. 3)TMTABB=TAB(14,14,2)
      PRINT 267, KZTB, KZRW, TAB(KZTB,KZRW,2),TMTABB
  267 FORMAT(7X,'GUIDANCE B :',T23,I2,T29,I2,T34,F8.3,T44,F7.3)
C
C         FIRE CONTROL
      KZTB = TAB(20,4,1)
      KZRW = TAB(20,4,2)
      PRINT 268, KZTB, KZRW, TAB(KZTB,KZRW,2),TAB(14,11,2)
  268 FORMAT(5X,'FIRE CONTROL :',T23,I2,T29,I2,T34,F8.3,T44,F7.3)
C
C         LAUNCHER
      KZTB = TAB(20,3,1)
      KZRW = TAB(20,3,2)
      PRINT 269, KZTB, KZRW, TAB(KZTB,KZRW,2),TAB(14,8,2)
  269 FORMAT(9X,'LAUNCHER :',T23,I2,T29,I2,T34,F8.3,T44,F7.3)
                                         PAUSE ';PRESS <<ENTER>> TO CONTINUE.'
 9800                                               RETURN
C
      END
C*************************************************************************
      SUBROUTINE RERUN
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &    ,     IFAC(48),    NT(48),     IRC(49),     IDDT(48)
     &    ,     IFD(48),    ITERM(48),     QTY(48),     CMAT(48)
      COMMON     RESS(48),    RESU(48),     RESB(48),     ITYPE(48)
     &    ,     PERC(48),    TDUE(48),     DEC(3),     Y(8)
     &    ,     IPRC(48),    RESC(48),     YF(48),     RESA(48)
     &    ,     TSTAR(48),    CL(6,5),     ALN(48),     CT(48)
     &    ,     TFIN(48),    TMIN(48),     IQ(48),     CUN(30)
      COMMON     RESMIN(30),    RESMAX(30),     AK(30),     ISW(3)
     &    ,     ITAB(25),    NN(8),     N(8),     INDX(8)
     &    ,     RATIO(8),    AN(9),     ADJ(2,8),     PDR(6)
```

```
      &    ,    PER(6) ,   PVR (6) ,    CD,    ICD,    INA,INF,IDP
       COMMCN CPUMAX,CPUMIN, FCPMIN,X,NTEMP(25) ,COV
      &    ,    C10,    QQ(18)
      &    ,    IACT(48,2),    IPC(48,5)
      &    ,    TDS(48,2) ,    JMIN,    IC,    IFIG,    KDP,    MF
       COMMCN NC,ICOFG,KP,KKPP,ITEAM
       COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
      &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
C
       ALPHA=QQ(14)
       IDP=4
       JMIN = 0
       NTEMP(2)=4
C
       DO 1 I=1,48
       IF (I .GT. 3) GO TO 4
       DEC(I)=0.
       ISW(I)=0.
     4 IF (I .GT. 6) GO TO 2
       PVR(I)=0.
       PDR(I)=0.
       PER(I)=0.
       DO 6 J=1,5
     6 CL(I,J)=0.
C
     2 IF (I .GT. 8) GO TO 3
       CTC(I)=0.
       INDX(I)=0
       RATIO(I)=0.
       ADJ(1,I)=0.
       ADJ(2,I)=0.
       NN(I)=0
C
     3 IF (I .GT. 9) GO TO 5
       MIS(I)=0
       AN(I)=0.
     5 IF (I .GT. 18) GO TO 15
       QQ(I)=0.
C
    15 IF (I .GT. 25) GO TO 7
```

```
      QT(I) = 0.
      ITAB(I)=0
    7 IF (I .GT. 32) GO TO 8
      DD(I)=0.
C
    8 RESC(I)=0.
C
      DO 10 J=1,2
      TDS(I,J)=0.
   10 IACT(I,J)=0
    1 CONTINUE
C
      DO 20 I=1,25
      DO 20 J=1,20
      IF (J .NE. 17) GO TO 22
      IF (I. LT. 5) GO TO 20
   22 TAB (J,I,3)=0.
   20 CONTINUE
C
      QC(14)=ALPHA
      RETURN
      END
C *************************************************************************
      SUBROUTINE STORE
C
      INTEGER IN/5/,DATFIL/9/
      COMMON/STORED/ STAT(5),NDUMP(5,5),ADUMP(5,28),DPCCST,MAXDP
      REAL*8 STAT
      REAL*8 FINAL/'FINAL'/,INTEN/'PROPOSED'/,BLK/'        '/
      REAL*8 NULL/'        '/
10    REWIND DATFIL
      DO 25 JA=3,5
      IF(STAT(JA).EQ.NULL)STAT(JA)=BLK
      WRITE(DATFIL,35)(NDUMP(JA,K),K=1,5),(ADUMP(JA,
     *K),K=1,28),STAT(JA)
25    CONTINUE
35    FORMAT(2I2,1X,3I1,2X,8F7.2,/,3X,9F7.2,/,3X,6F7.2,/,3X,5F7.2,2X,A8)
      RETURN
      END
C *************************************************************************
```

```
      SUBROUTINE FINISH
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &   ,    IFAC(48),   NT(48),    IRC(49),    IDDT(48)
     &   ,    IFD(48),   ITERM(48),   QTY(48),   CMAT(48)
      COMMON   RESS(48),   RESU(48),   RESB(48),   ITYPE(48)
     &   ,    PERC(48),   TDUE(48),   DEC(3),   Y(8)
     &   ,    IPRC(48),   RESC(48),   YF(48),   RESA(48)
     &   ,    TSTAR(48),   CL(6,5),   ALN(48),   CT(48)
     &   ,    TFIN(48),   TMIN(48),   IQ(48),   CUN(30)
      COMMON   RESMIN(30),   RESMAX(30),   AK(30),   ISW(3)
     &   ,    ITAB(25),   NN(8),   N(8),   INDX(8)
     &   ,    RATIO(8),   AN(9),   ADJ(2,8),   PDR(6)
     &   ,    PER(6),   PVR(6),   CD,   ICD,   INA,INF,IDP
      COMMON CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
     &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
      COMMON   COSTP2,   COSTOT,   ITCT,   ALPHA,   PRIOR
     &   ,    TRADEP,   TF,   WA,   IACT(48,2),   IPC(48,5)
     &   ,    TDS(48,2),   JMIN,   IC,   IFIG,   KDP,   MR
     &   ,    NC,ICOFG,KP,KKPP,ITEAM
      COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
      COMMON/STORED/ STAT(5),NDUMP(5,5),ADUMP(5,28),DPCCST,MAXDP
          COMMON/CODE/ ICODE,FLAG,ITEM,STORFL
          REAL*8 ICODE,FLAG,ITEM,FL21/'TEAM21'/
      REAL*8 STAT,PROP
      REAL*8 FINAL/'FINAL'/,INTEN/'PROPOSED'/,BLK/'        '/
      REAL*8 INCCD,CCNT/'CONT'/,NULL/'        '/,EASY/'E       '/
      INTEGER IN/5/,DATFIL/9/
                                             CALL ERRSET(218,1,-1,1,1)
10                    PROP=INTEN
            IF(STAT(KDP).EQ.INTEN) PROP=FINAL
            IF(STAT(KDP).EQ.BLK) PROP=INTEN
            IF(STAT(KDP).EQ.NULL) PROP=INTEN
417                                 CALL FRTCMS('CLRSCRN ')
      PRINT 415,ITEAM,KDP,STAT(KDP)
415   FORMAT(1X,'TEAM ',I2,', IN DP-',I1,', IS IN STATUS ',A8, '.')
            IF(MR.NE.2)GOTO 410
      PRINT 430
430   FORMAT(1X,
```

```
     * 'WHAT DP AND WHAT STATUS DO YOU WANT; EX: 3 FINAL; 4 PROPOSED ?')
       READ(IN,416,ERR=420,END=420)KDP,PROP
416    FORMAT(1I1,1X,1A8)
                                             IF(PROP.EQ.INTEN)STAT(KDP)=BLK
                                             IF(PROP.EQ.FINAL)STAT(KDP)=INTEN
                   IF(KDP.EQ.EASY) RETURN
                   IF(KDP.LT.3.OR.KDP.GT.5)GOTO 417
             IF(PROP.EQ.INTEN.OR.PROP.EQ.FINAL)CALL FRICMS('CLRSCRN ')
                   IF(PROP.EQ.INTEN.OR.PROP.EQ.FINAL)GOTO 410
                                                       GOTO 417
420                                                    RETURN
410    CONTINUE
                                         PRINT 85,KDP,NDUMP(KDP,3),
     *                                 ADUMP(KDP,13),NDUMP(KDP,4),ADUMP(KDP,14)
     *                                 ,NDUMP(KDP,5),ADUMP(KDP,15),ADUMP(KDP,1)
     *                                   ,ADUMP(KDP,16)
                                         PRINT 90,ADUMP(KDP,2),ADUMP(KDP,17),
     }                                   ADUMP(KDP,3),ADUMP(KDP,18),
     }                                   ADUMP(KDP,4),ADUMP(KDP,19),
     }                                   ADUMP(KDP,5),ADUMP(KDP,20),
     }                                   ADUMP(KDP,6),ADUMP(KDP,21),
     }                                   ADUMP(KDP,7),ADUMP(KDP,22)
                                         PRINT 110,ADUMP(KDP,8),ADUMP(KDP,23),
     }                                   ADUMP(KDP,9),ADUMP(KDP,24),
     }                                   ADUMP(KDP,10),ADUMP(KDP,25),
     }                                   ADUMP(KDP,11),ADUMP(KDP,26),
     }                                   ADUMP(KDP,12),ADUMP(KDP,27)
              IF(STAT(KDP).EQ.BLK.OR.STAT(KDP).EQ.NULL)PRINT 36,INTEN,KDP
36     FORMAT(' ****DO YOU WISH TO ENTER THIS DATA AS YOUR ',A8,' DP-',
     *I2,' DECISION?****')
                                         IF(STAT(KDP).EQ.INTEN)PRINT 37,KDP
37     FORMAT(1X,                     32('*'),'IMPORTANT',32('*'),/,1X,5('*'),'
     *YOU MUST CHECK THE ABOVE ENTRIES WITH THE APPROVED FINAL DP-',I1,'
     * .',5('*')                      )
                                         PRINT 38
38     FORMAT(4X,    'ENTER YOUR TEAM SECURITY CODE IF YOU WISH TO COMMIT
     *TO A DECISION;',/,30X,'OR "CONT".')
                                         READ(IN,125,ERR=10,END=10)INCOD
125                                      FORMAT(A8)
             IF(INCOD.EQ.CONT)RETURN
```

```
               IF(INCCD.EQ.EASY)RETURN
               IF(INCCD.NE.ICODE)GOTO 300
                          STAT(KDP)=PROP
          CALL STORE
310                                    CALL FRTCMS('CLRSCRN ')
                                       PRINT 105, ITEAM,KDP,ERCP
                                       PRINT 85,KDP,NDUMP(KDP,3),
     *                                ADUMP(KDP,13),NDUMP(KDP,4),ADUMP(KDP,14)
     *                                ,NDUMP(KDP,5),ADUMP(KDP,15),ADUMP(KDP,1)
     *                                 ,ADUMP(KDP,16)
                                       PRINT 90,ADUMP(KDP,2),ADUMP(KDP,17),
     )                                 ADUMP(KDP,3),ADUMP(KDP,18),
     )                                 ADUMP(KDP,4),ADUMP(KDP,19),
     )                                 ADUMP(KDP,5),ADUMP(KDP,20),
     )                                 ADUMP(KDP,6),ADUMP(KDP,21),
     )                                 ADUMP(KDP,7),ADUMP(KDP,22)
                                       PRINT 110,ADUMP(KDP,8),ADUMP(KDP,23),
     )                                 ADUMP(KDP,9),ADUMP(KDP,24),
     )                                 ADUMP(KDP,10),ADUMP(KDP,25),
     )                                 ADUMP(KDP,11),ADUMP(KDP,26),
     )                                 ADUMP(KDP,12),ADUMP(KDP,27)
85        FORMAT(' NO.*ITEM******CURRENT VALUE @ DP-',I1,4X,'NO.*ITEM*******
     ******CURRENT VALUE',/,'  1. CONTRACTOR                :',2X,I1, 8X,'16
     *. GUIDANCE QUAL TESTS    :',F6.2,/,'  2. GUIDANCE APPROACH     :',2
     *X,I1, 8X,'17. FLIGHT TESTS',10X,':',F6.2,/,'  3. GUIDANCE CONFIGUR
     *ATION:',2X,I1, 8X,'18. MAXIMUM COST             M$:',F6.2,/,'  4. MAINT
     *AINABILITY ENG M$:',F6.2,  5X,'19. MINIMUM COST            M$:',F6.2)
90        FORMAT('  5. VALUE ENG                 M$:',F6.2,4X,
     *' 20. MAX COST INCENTIVE    %:',F6.2
     *,/,'  6. PARALLEL DEVELOP     M$:',F6.2,4X,
     *' 21. LATEST WEEK             :',F6.2
     *,/,'  7. MOTOR RELIABILITY     %:',F6.2,4X,
     *' 22. EARLIEST WEEK           :',F6.2
     *,/,'  8. AIRFRAME RELIABILITY %:',F6.2,4X,
     *' 23. MAX DELIVER INCENTIVE%:',F6.2
     *,/,'  9. LAUNCHER RELIABILITY %:',F6.2,4X,
     *' 24. MAXIMUM RELIABILITY   %:',F6.2
     *,/,' 10. FIRE CONTROL ERROR YDS:',F6.2,4X,
     *' 25. MINIMUM RELIABILITY   %:',F6.2)
110       FORMAT(' 11. GUIDANCE ERROR        YDS:',F6.2,4X,
```

```
     *' 26. MAX RELIABIL INCENTIV%:',F6.2
     *,/,' 12. MOTOR QUAL TESTS      :',F6.2,4X,
     *' 27. MAXIMUM ERROR        YDS:',F6.2
     *,/,' 13. AIRFRAME QUAL TESTS   :',F6.2,4X,
     *' 28. MINIMUM ERROR        YDS:',F6.2
     *,/,' 14. LAUNCHER QUAL TESTS   :',F6.2,4X,
     *' 29. MAX ERROR INCENTIVE  %:',F6.2
     *,/,' 15. FIRE CONTROL QUAL TSTS:',F6.2,4X,
     *' 30. WEEK FOR IOT 10       :',F6.2)
105   FORMAT(14X,'******THE FOLLOWING PARAMETERS WERE ACCEPTED******',
     */,14X,'AS TEAM ',I2,' DP-',I2,1X,A8,' INPUT TO THE CONTRACTOR.')
        IF(MR.EQ.0) PRINT 425
425   FORMAT(
     *14X,'*****SEE YOUR MONITOR IF YOU HAVE A PROBLEM.******')
                                        PAUSE ';PRESS <<ENTER>> TO CONTINUE.'
C                          PROP=INTEN
      IF(STAT(KDP).EQ.FINAL)KDP=KDP+1
      RETURN
300                                     PRINT 305
305   FORMAT(' *****WARNING*****YOUR CODE WAS NOT ENTERED CORRECTLY.')
                                        PAUSE ';PRESS <<ENTER>>'
      RETURN
      END
C***************************************************************************
      SUBROUTINE PLTSCH
C
      COMMON/PLOTVA/XBASE,YCURVE
                               DIMENSION XBASE(12),YCURVE(12,12)
                               REAL RBLK/' '/
               INTEGER HORIZ/' '/,CHARA(12),J/'J'/,K/'K'/,M/'M'/,N/'N'/
     *,A/'A'/,B/'B'/,C/'C'/,D/'D'/,E/'E'/,F/'F'/,G/'G'/,H/'H'/,I/'I'/
       INTEGER QUES/'?'/,L/'L'/,BLK/' '/,PRD/'.'/,LINE(55,22),VERT/'|'/
     *,CCM/','/,AST/'*'/,PLUS/'+'/,DASH/'-'/,QUOTE/'"'/,CENT/'¢'/,DOL/'$
     *'/,POUND/'#'/,AT/'@'/,LESS/'<'/,GREAT/'>'/,COLON/':'/,SEMICO/';'/
     *,PRTPCT/'%'/,AMPER/'&'/,R/'R'/,T/'T'/
                               INTEGER SPRIT/' '/,CHARKY(20)
                               REAL*8 CHARDF(20)
                                        REAL*8 DVCOST/'DEV COST'/
                                        REAL*8 TSTCMP/'TESTCOMP'/
                                        REAL*8 RLIBIL/'RELIABIL'/
```

```
                                  REAL*8 ACRACY/'ACCURACY'/
                                  REAL*8 TALFEE/'TOTLFEE%'/
                                  REAL*8 TLCOST/'TCTICOST'/
                                  REAL*8 EFINDX/'EFFINDEX'/
                                  REAL*8 MULPTS/'MULTPTS'/
                                         CHARDF(4)= LVCOST
                                         CHARDF(6)= TSTCMP
                                         CHARDF(8)= RLIBIL
                                         CHARDF(10)= ACRACY
                                         CHARDF(12)= TALFEE
                                         CHARDF(14)= TLCOST
                                         CHARDF(16)= EFINDX
                                         CHARDF(18)= MULPTS
C ZERCIZE 'ENTRY ROUTINE' FLAGS AND PARAMETERS
                                            PCTFLG=0.0
                                            YMX=99999.9
                                            YMN=0.0
C ALL ENTRY ROUTINES @ THE END OF PLTSCH JUMP IN HERE
10                                          CONTINUE
C INITIALIZE ALL X AND Y ARRAYS THAT ARE NOT DEFINED BY THE INPUT DATA
C PROVIDE GRAPH FORMAT FOR GRID SYSTEM
                        DO 40 INCRE=1,11
                             DC 40 JNCRE=1,5
                                  DO 40 KNCREM=1,10
                                       DO 40 LNCREM=1,2
                                            INCREM=(INCRE-1)*5+JNCRE
                                            JNCREM=(KNCREM-1)*2+LNCREM
                                            LINE(INCREM,JNCREM)=BLK
                                 IF(LNCREM.EQ.1) LINE(INCREM,JNCREM)=PRD
                                 IF(JNCRE.EQ.1) LINE(INCREM,JNCREM)=PRD
                                 IF(INCREM.EQ.1) LINE(INCREM,JNCREM)=VERT
                 IF(LNCREM.EQ.1.AND.INCREM.EQ.1) LINE(INCREM,JNCREM)=PLUS
40                      CONTINUE
                        DO 50 KNCREM=1,11
                             DO 50 LNCREM=1,5
                                  JNCREM=(KNCREM-1)*5+LNCREM
                                  LINE(JNCREM,21)=HORIZ
                                 IF(LNCREM.EQ.1) LINE(JNCREM,21)=VERT
50                      CONTINUE
C                                           XMAX=0.0
```

```
C                                          XMIN=0.0
                                           LINE(51,21)=VERT
                   IF(YMX.NE.99999.9)GOTO 62
C DETERMINE X ANC Y SET SIZE, MIN, AND MAX FOR SCALE GENERATION
                                           YMAX=0.0
                                           YMIN=0.0
                   IF(PCTFLG.NE.0.0)GOTO 63
                        DO 60 INCRE=1,11
                        DO 60 JNCRE=1,11
                                        IF(YCURVE(JNCRE,INCRE).GT.YMAX)
     *                                     YMAX=YCURVE(JNCRE,INCRE)
C                                       IF(YCURVE(JNCRE,INCRE).LT.YMIN)
C    *                                     YMIN=YCURVE(JNCRE,INCRE)
C                                       IF(XBASE(JNCRE).GT.XMAX)
C    *                                     XMAX=XBASE(JNCRE)
C                                       IF(XBASE(JNCRE).LT.XMIN)
C    *                                     XMIN=XBASE(JNCRE)
60                      CONTINUE
                                           GOTO 61
C THE NEXT TWO LINES SET THE Y-AXIS VALUES FOR THE SCALED ROUTINE
62                      CONTINUE
      IF(YMX.NE.99999.9)YMIN=YMN
      IF(YMX.NE.99999.9)YMAX=YMX
C APPROXIMATE THE AXIS SCALES.
61                                         YSIZE=ABS(YMAX-YMIN)
                                           IF(YSIZE.EQ.0.0)YSIZE=1.0
                                           YSCALE=(YSIZE/20.0)
C                                          XSIZE=ABS(XMAX-XMIN)
C                                          IF(XSIZE.EQ.0.0)XSIZE=1.0
C                                          XSCALE=(XSIZE/20.0)
                                                GOTC 63
C ESTAELISH THE X ANC Y SCALES AS STANDARD MODULUS'.
C64     YLOG=ALOG10(YSCALE)
C       IYLOG=YLOG
C       DYLOG=ABS(YLOG-IYLOG)
C       DO 74 JITIG=1,20
C       BIT=(21.0-JITIG)/2.0
C       SIG=YLOG/(ABS(YLOG))
C       IF(DYLOG.GE.AICG10(BIT))GO TO 76
C74     CCNTINUE
```

```
C76    YSCALE=10**((IYLOG+ALOG10(BIT))*SIG)
C      IF(YSCALE.LT.0.0005) YSCALE=0.0005
C MAKE THE AXIS MIN AND MAX INTEGER MULTIPLES OF THE SCALE
C                 YMIN=SIGN(INT(ABS(YMIN/YSCALE))+1.0,YMIN)
C                 YMAX=SIGN(INT(ABS(YMAX/YSCALE))+1.0,YMAX)
63    IF(PCTFLG.NE.0.0)YSCALE=0.5
C                          DEVELOP LINE AND COLUMN MATRIX OF CHARACTERS
C                          'CHARA' CHARACTERS REPRESENT DATA POINTS:
C                    = COST DATA
C                  T = FLIGHT TEST COMPLETION DATE DATA
C                  R = RELIABILITY DATA
C                  @ = ACCURACY DATA
C                  % = TOTAL FEE PERCENTAGE
C                  $ = TOTAL COST
C                  * = EFFECTIVENESS INDEX
C                  & = MULTIPLE CURVE INTERSECTION
                                           CHARA(1)=CENT
                                           CHARA(2)=CENT
                                           CHARA(3)=T
                                           CHARA(4)=T
                                           CHARA(5)=R
                                           CHARA(6)=R
                                           CHARA(7)=AT
                                           CHARA(8)=AT
                                           CHARA(9)=PRTPCT
                                           CHARA(10)=DCL
                                           CHARA(11)=AST
C                                          CHARA(12)=AMPER
      DO 70 IYNM=1,12
           DO 70 IXNM=1,11
                DO 70 INERW=1,21
                                           YFAC=(21-INERW)*YSCALE+YMIN
                                           INROW=INERW+1
                     DO 70 LINECL=1,51
C     IF(ABS(YCURVE(IYNM,IXNM)-YFAC).LE.(0.5*YSCALE).AND.(IXNM-1)*5.EQ.(
C    *LINECL-1).AND.(LINE(LINECL,INROW).NE.BLK.OR.LINE(LINECL,INROW).NE.
C    *HORIZ.OR.LINE(LINECL,INROW).NE.VERT))
C    *LINE(LINECL,INROW)=AMPER
      IF(ABS(YCURVE(IYNM,IXNM)-YFAC).LE.0.5*YSCALE.AND.
     *(IXNM-1)*5.EQ.(LINECL-1).AND.LINE(LINECL,INROW).NE.AMPER)
```

```
     *LINE(LINECL,INROW)=CHARA(IYNM)
70    CONTINUE
                      CHARKY(4)=CENT
                      CHARKY(6)=T
                      CHARKY(8)=R
                      CHARKY(10)=AT
                      CHARKY(12)=PRTPCT
                      CHARKY(14)=DOL
                      CHARKY(16)=AST
                      CHARKY(18)=AMPER
            CALL FRTCMS('CLRSCRN ')
              DO 80 JNROW=1,10
                      JROW=(JNROW-1)*2
                      KROW=JROW+1
                      YFACT=(21-KROW)*YSCALE+YMIN
                PRINT 85,YFACT,(LINE(INCREM,KROW),INCREM=1,51)
85    FORMAT(F8.2,T10,51A1)
                      KROW=KROW+1
      IF(KROW.GE.4.AND.KROW.LE.18)PRINT 89,(LINE(INCREM,KROW),INCREM=1,5
     *1),CHARKY(KROW),CHARDF(KROW)
89    FORMAT(T10,51A1,2X,A1,' = ',A8)
      IF(KROW.LT.4.OR.KROW.GT.18)PRINT 90,(LINE(INCREM,KROW),INCREM=1,51
     *)
90    FORMAT(T10,51A1)
80    CONTINUE
                PRINT 95,YMIN,(LINE(KNCREM,21),KNCREM=1,51)
95    FORMAT(F8.2,T10,51A1,3X,'Y < 15 ====>%FEE')
      IF(XBASE(11).GE.100.0)GOTO 147
      IF(XBASE(11).GE.10.0)GOTO 148
140   PRINT 145,(XBASE(INCRE),INCRE=1,11)
145   FORMAT(T8,11F5.2,/)
C                     PAUSE ';PRESS <<ENTER>> TO CONTINUE.'
                              RETURN
147   PRINT 149,(XBASE(INCRE),INCRE=1,11)
149   FORMAT(T6,11F5.0,/)
C                     PAUSE ';PRESS <<ENTER>> TO CONTINUE.'
                              RETURN
148   PRINT 150,(XBASE(INCRE),INCRE=1,11)
150   FORMAT(T7,11F5.1,/)
C                     PAUSE ';PRESS <<ENTER>> TO CONTINUE.'
```

```
                                                  RETURN
C...........GENERATE ENTRY CONDITIONS FOR THE FOLLOWING 'ENTRY ROUTINES'
      ENTRY PLTSCL(YMN,YMX)
                                                  PCTFLG=0.0
                              GOTO 10
      ENTRY PLTPCT(PCTFLG)
                                                  YMN=0.0
                                                  YMX=99999.9
                              GOTC 10
      END
C*****=============================================================*****
C
      SUBROUTINE FSEFF(TOC,FINDEX)
C
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &    ,     IFAC(48),    NT(48),     IRC(49),     IDDT(48)
     &    ,     IFD(48),   ITERM(48),     QTY(48),     CMAT(48)
      COMMON     RESS(48),     RESU(48),     RESB(48),     ITYPE(48)
     &    ,     PERC(48),    TDUE(48),     DEC(3),      Y(8)
     &    ,     IPRC(48),    RESC(48),     YF(48),     RESA(48)
     &    ,     TSTAR(48),    CL(6,5),     ALN(48),     CT(48)
     &    ,     TFIN(48),    TMIN(48),     IQ(48),     CUN(30)
      COMMCN     RESMIN(30),     RESMAX(30),     AK(30),     ISW(3)
     &    ,     ITAB(25),    NN(8),     N(8),     INDX(8)
     &    ,     RATIO(8),    AN(9),     ADJ(2,8),     PDR(6)
     &    ,     PER(6),    PVR(6),    CD,    ICD,    INA,INF,IDP
      COMMCN CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
     &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
      COMMON     COSTP2,     COSTOT,     ITCT,     ALPHA,     PRICR
     &    ,     TRADEP,    TF,    WA,    IACT(48,2),     IPC(48,5)
     &    ,     TDS(48,2),     JMIN,    IC,    IFIG,    KDP,    MR
     &    ,     NC,ICOFG,KP,KKPP,ITEAM
      COMMCN RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
      REAL XRAY(20),ZEBRA(20),AFSE/65./,THAVL/85./
C
C
C         EXPECTED AVAILABILITY= THEORETICAL AVAIL/(1+AVAILBILITY FACTOR)
C          AVAILABILITY FACTCR(MAINTAINABILITY FUNDS)=.29*EXP(-.44*FUNDS)
```

```
C         THE AVFCTR FORMULA IS FROM AN EXPONENTIAL CURVE FITTING SOLU-
C             TION OBTAINED ON AN HP-41CV STATISTICS PACK MODULE. THE
C             REGRESSION (R**2) COEFFICIENT ACHIEVED WAS 1.00.
      AVFCTR=0.29*(1/(EXP(0.44*Y(1))))
      AVAIL=THAVL/(1+AVFCTR)
      IF(AVAIL.LT.0.)AVAIL=30.
      IF(AVAIL.GT.100.)AVAIL=100.
C                 AVAIL = MTBF * 100.0 / (MTBF + MTTR)
10       SSPK = .33375+AVAIL*(1.8523*(1/(EXP(.00634*Y(4)))))
      IF(SSPK.GT.1.)SSPK=1.
C         START DATE FOR PRODUCTION OF BLOCK 1 LOTS IS COMPLETION OF THE
C             FLIGHT TESTS, Y(2), IN FLY-BEFORE-BUY. IT IS DELAYED BY 3
C             MONTHS FOR
      DELAY=8.0*(1.0+SIN(MFTDOL-1.1)*3.141593/2.0)
      START1=(INT((Y(2)+DELAY)/13.0)+1.0)*13.0
95       FSE = 100.0 - (100.0 -SSPK)**2.0/100.0
      IF(FSE.GT.85.)FSE=85.
         CALL SCHEDU(FSE,AFSE,START1)
C        EINDEX = 53.3010+1.44635*AFSE-0.953836*(TOC+40.)
          EINDEX = (0.993-ALOG((40.+TOC)/AFSE))/0.01066
                                            RETURN
      END
C.........................................................................
      SUBROUTINE SCHEDU(FSE,AFSE,START1)
         REAL    BIGGER,AFSE,SUMFSE,ZFSE,XRAY(20),ZEBRA(20),FSE
         DATA    XRAY/50.0,50.0,50.0,40.0,30.0,20.0,14*0.0/
C                                 (GENERATE ZEBRA PREDICTED FSE ARRAY)
         DO 50 I=1,20
            ZEBRA(I)=0.0
            ZFSE = 0.0
            NWEEK = 183 + 26 * I
            IF(START1.GT.NWEEK) GOTO 40
               DO 10 J=1,10
                  ZWEEK = START1 +13.0* (J-1)
                  IF (ZWEEK.LE.NWEEK) ZFSE = ZFSE + FSE/10.0
  10           CONTINUE
  40        CONTINUE
            IF(ZFSE.GT.80.0) ZFSE = 80.0
            ZEBRA(I) = ZFSE
  50     CONTINUE
```

```
C                               (PICK LARGER FSE, THEN CALCULATE AVERAGE)
          SUMFSE = 0.0
          DO 80 I=1,20
                BIGGER = XRAY(I)
             IF(XRAY(I).LT.ZEBRA(I))
     *          BIGGER = ZEBRA(I)
             SUMFSE = SUMFSE + BIGGER
   80     CONTINUE
          AFSE = SUMFSE / 20.0
          RETURN
       END
C*************************************************************************
C
       SUBROUTINE SNSITV
C
       COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &    ,    IFAC(48),    NT(48),     IRC(49),     IDDT(48)
     &    ,    IFD(48),   ITERM(48),    QTY(48),     CMAT(48)
       COMMON   RESS(48),    RESU(48),     RESB(48),     ITYPE(48)
     &    ,    PERC(48),    TDUE(48),     DEC(3),     Y(8)
     &    ,    IPRC(48),    RESC(48),     YF(48),     RESA(48)
     &    ,    TSTAR(48),    CL(6,5),     ALN(48),     CT(48)
     &    ,    TFIN(48),    TMIN(48),     IQ(48),     CUN(30)
       COMMON   RESMIN(30),    RESMAX(30),    AK(30),    ISW(3)
     &    ,    ITAB(25),    NN(8),    N(8),    INDX(8)
     &    ,    RATIO(8),    AN(9),    ADJ(2,8),    PDR(6)
     &    ,    PER(6),    PVR(6),    CD,    ICD,    INA,INF,IDP
       COMMON CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV
     &    ,    C10,    CQ(18)
     &    ,    IACT(48,2),    IPC(48,5)
     &    ,    TDS(48,2),    JMIN,    IC,    IFIG,    KDP,    MF
       COMMON NC,ICOFG,KP,KKPP,ITEAM
       COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TDMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
C
       COMMON/PLOTVA/SAVSEN,SAVDC
       COMMON/STORED/ STAT(5),NDUMP(5,5),ADUMP(5,28),DPCCST,MAXDP
       REAL XFACT(11),RBLK/' '/,YFEE(8),FRATIO(8),FYC(5)
C THE FOLLOWING SETS OF REAL DATA CREATE LABELS FOR THE TABLES.
       REAL*8 AREA1,AREA2,AREA3
```

```
      REAL*8 LABEL,MAX/'MAXIMUM'/,MIN/'MINIMUM'/,LATE/'LATE '/,
     *EARLY/'EARLY'/,HIGH/' HIGH '/,LOW/'LOW    '/,GREAT/'GREAT'/
     *,LESS/'LESS '/,BLK/'        '/
      REAL*8 GUIDE/'GUIDANCE'/,DEVEL/'DEVELOPM'/,FLIGHT/'FLIGHT'/
      REAL*8 TEST/'  TESTS '/,ACCUR/'ACCURACY'/,ENGIN/'ENGINEER'/
      REAL*8 RELIA/'RELIABIL'/,MOTOR/'  MOTOR '/,AIRFR/'AIRFRAME'/
      REAL*8 MAINT/'MAINTAIN'/,PARAL/'PARALLEL'/,COST/'  COST  '/
      REAL*8 DELIV/'DELIVERY'/,FUNDS/'  FUNDS '/,VALUE/'  VALUE '/
      REAL*8 LAUNC/'LAUNCHER'/,FIREC/'FIRECONT'/,UNDRL/'________'
     */,QUAL/'  QUAL  '/,INCEN/'INCENTIV'/,TESTC/'TESTCCMP'/
      REAL LOWER,SAVSEN(12),SAVDC(12,12)
          INTEGER ANS,ANSW,ANO/'N'/,YES/'Y'/,E/'E'/,S/'S'/
          INTEGER IN/5/,SENSTY,CKPLT,ISAVD2(11),ISAVD4(11)
                                            LABEL=BLK
                                            CALL ERRSET(218,1,-1,1,1)
                                            ISEN=0
                                            SELX=0.
                                            EIOPT=0.
                                   DO 10 ITIG=1,11
                                            SAVSEN(ITIG)=0.
                                   DO 10 JITIG=1,11
                                            SAVDC(ITIG,JITIG)=0.
10                                 CONTINUE
20                                          CALL FRTCMS('CLRSCRN ')
                                            PRINT 35
35    FORMAT(' *****SELECT THE ITEM WHICH IS TO BE SENSITIVITY ANALYZED.
     ******',/)
                                            PRINT 15
15    FORMAT(
     *'  1. MAINTAINABILITY ENG COST  13. GUIDANCE QUAL TESTS',/
     *,'  2. VALUE ENGINEERING COST    14. FLIGHT TESTS ',/,
     *'  3. PARALLEL GUIDANCE COST    15. DEVELOPMENT COST RANGE',/
     *,'  4. MOTOR RELIABILITY RANGE   16. DEV COST INCENTIVE %',/
     *,'  5. AIRFRAME RELIABLE RANGE   17. END WK OF FLIGHT TEST RANGE',
     */,'  6. LAUNCHER/GSE RELIABILITY  18. FLIGHT TEST CCMP INCENTIVE')
      PRINT 21
21    FORMAT(
     *' 7. FIRE CONTROL IMPACT ERROR 19. SYSTEM RELIABILITY RANGE',/,
     *' 8. GUIDANCE IMPACT ERROR     20. SYS RELIABILITY INCENTIVE %',/
     *,'  9. MOTOR QUAL TESTS         21. IMPACT ERROR RANGE',/,
```

```
     *' 10. AIRFRAME QUAL TESTS      22. IMPACT ACCURACY INCENTIVE',/,
     *' 11. LAUNCHER/GSE QUAL TESTS   23. DEPLOYMENT DATE RANGE',/,
     *' 12. FIRE CONTROL QUAL TESTS ',/,
     *20X,'*******0. NONE*******')
                                         READ(IN,*,ERR=20,END=20)SENSTY
                                 IF(SENSTY.EQ.0)RETURN
C
                                           CALL FRTCMS('CLRSCRN ')
      DO 2600 ISEN=1,11
                                           RATIO2=1.
                                           CALL ZRW
            NTEMP(1)=KDP
      DO 150 IVER2=3,KDP
                        CALL GET(IVER2)
105   GOTO(200,300,400,500,600,700,800,900,1000,1100,1200,1300,1400,
     *1500,1600,1700,1800,1900,2000,2100,2200,2300,2400),
     *SENSTY
            GOTO 20
200   CONTINUE
            IF(ISEN.GT.1)GOTO 220
      AREA1=MAINT
      AREA2=ENGIN
      AREA3=FUNDS
            PRINT 210
210   FORMAT(' INPUT THE VALUES TO VARY MAINT ENG FUNDS BETWEEN.',
     */,' WHAT IS THE LOWER VALUE: >1.5 ?')
            READ(IN,*,ERR=200,END=200)LOWER
      IF(LOWER.LE.1.5)LOWER=1.5
            PRINT 215
215   FORMAT(' WHAT IS THE UPPER VALUE: <4.0 ?')
            READ(IN,*,ERR=200,END=200)UPPER
      IF(UPPER.GT.4.0)UPPER=4.0
                   IF(LOWER.GT.UPPER)GOTO 200
               RANGE=(ABS(UPPER-LOWER))*.1
220      CTC(8)=LOWER+RANGE*(ISEN-1)
               IF(CTC(8).GT.UPPER)GOTO 2620
         SENVAR=CTC(8)
      GOTO 2440
300   CONTINUE
            IF(ISEN.GT.1)GOTO 320
```

```
      AREA1=VALUE
      AREA2=ENGIN
      AREA3=FUNDS
            PRINT 310
310   FORMAT(' INPUT THE VALUES TO VARY VALUE ENG FUNDS BETWEEN.',
     */,' WHAT IS THE LOWER VALUE, MUST BE 1.75 TO 2.2 MAX ?')
            READ(IN,*,ERR=300,END=300)LOWER
               IF(LOWER.LE.0.5)LOWER=0.5
            PRINT 315
315   FORMAT(' WHAT IS THE UPPER VALUE, MUST BE LESS THAN 2.2?')
            READ(IN,*,ERR=300,END=300)UPPER
C              IF(UPPER.GT.3.0)UPPER=3.0
                   IF(LOWER.GT.UPPER)GOTO 300
               RANGE=(ABS(UPPER-LOWER))*.1
320      CTC(7)=LOWER+RANGE*(ISEN-1)
               IF(CTC(7).GT.UPPER)GOTO 2620
         SENVAR=CTC(7)
      GOTO 2440
400   CONTINUE
            IF(KDP.GT.3.OR.MIS(6).EQ.2.OR.MIS(6).EQ.4)GOTO 20
            IF(ISEN.GT.1)GOTO 420
      AREA1=PARAL
      AREA2=DEVEL
      AREA3=FUNDS
            PRINT 410
410   FORMAT(' INPUT THE VALUES TO VARY PARALLEL DEVELOPMENT FUNDS.'
     *,/,' WHAT IS THE LOWER VALUE?')
           READ(IN,*,ERR=400,END=400)LOWER
               IF(LOWER.LE.0)LOWER=.01
           PRINT 415
415   FORMAT(' WHAT IS THE UPPER VALUE?')
           READ(IN,*,ERR=400,END=400)UPPER
               IF(LOWER.GT.UPPER)GOTO 400
               RANGE=(ABS(UPPER-LOWER))*.1
420      CTC(5)=LOWER+RANGE*(ISEN-1)
               IF(CTC(5).GT.UPPER)GOTO 2620
         SENVAR=CTC(5)
      GOTO 2440
500   CONTINUE
            IF(ISEN.GT.1)GOTO 520
```

```
      AREA1=MOTOR
      AREA2=RELIA
      AREA3=UNDRL
            PRINT 510
510   FORMAT(' INPUT THE VALUES TO VARY MOTOR RELIABILITY BETWEEN. '
     *,/,' WHAT IS THE LOWER VALUE?')
            READ(IN,*,ERR=500,END=500)LOWER
               IF(LOWER.LE.0)LOWER=.01
            PRINT 515
515   FORMAT(' WHAT IS THE UPPER VALUE?')
            READ(IN,*,ERR=500,END=500)UPPER
                        IF(UPPER.GT.100.)UPPER=100.
                   IF(LOWER.GT.UPPER)GOTO 500
               RANGE=(ABS(UPPER-LOWER))*.1
520      YC(1)=LOWER+RANGE*(ISEN-1)
               IF(YC(1).GT.UPPER)GOTO 2620
         SENVAR=YC(1)
      GOTO 2440
600   CONTINUE
             IF(ISEN.GT.1)GOTO 620
      AREA1=AIRFR
      AREA2=RELIA
      AREA3=UNDRL
             PRINT 610
610   FORMAT(' INPUT THE VALUES TO VARY AIRFRAME RELIABILITY BETWEEN. '
     *,/,' WHAT IS THE LOWER VALUE?')
            READ(IN,*,ERR=600,END=600)LOWER
               IF(LOWER.LE.0)LOWER=.01
            PRINT 615
615   FORMAT(' WHAT IS THE UPPER VALUE?')
            READ(IN,*,ERR=600,END=600)UPPER
                        IF(UPPER.GT.100.)UPPER=100.
                   IF(LOWER.GT.UPPER)GOTO 600
               RANGE=(ABS(UPPER-LOWER))*.1
620      YC(2)=LOWER+RANGE*(ISEN-1)
               IF(YC(2).GT.UPPER)GOTO 2620
         IF(YC(2).GT.100..OR.YC(2).LE.0.)GOTO 2620
         SENVAR=YC(2)
      GOTO 2440
700   CONTINUE
```

```
            IF(ISEN.GT.1)GOTO 720
      AREA1=LAUNC
      AREA2=RELIA
      AREA3=UNDRL
            PRINT 710
710   FORMAT(' INPUT THE VALUES TO VARY LAUNCHER RELIABILITY BETWEEN.'
     *,/,' WHAT IS THE LOWER VALUE?')
            READ(IN,*,ERR=700,END=700)LOWER
               IF(LOWER.LE.0)LOWER=.01
            PRINT 715
715   FORMAT(' WHAT IS THE UPPER VALUE?')
            READ(IN,*,ERR=700,END=700)UPPER
                        IF(UPPER.GT.100.)UPPER=100.
                   IF(LOWER.GT.UPPER)GOTO 700
               RANGE=(ABS(UPPER-LOWER))
720      YC(3)=LOWER+RANGE*.1*(ISEN-1)
                   IF(YC(3).GT.UPPER)GOTO 2620
         IF(YC(3).GT.100..OR.YC(3).LE.0.)GOTO 2620
         SENVAR=YC(3)
      GOTO 2440
800   CONTINUE
            IF(ISEN.GT.1)GOTO 820
      AREA1=FIREC
      AREA2=ACCUR
      AREA3=UNDRL
            PRINT 810
810   FORMAT(' INPUT THE VALUES TO VARY FIRE CONTROL IMPACT BETWEEN.'
     *,/,' WHAT IS THE LOWER VALUE, MUST BE GREATER THAN 0?')
            READ(IN,*,ERR=800,END=800)LOWER
               IF(LOWER.LE.0)LOWER=.01
            PRINT 815
815   FORMAT(' WHAT IS THE UPPER VALUE, MUST BE LESS THAN 100?')
            READ(IN,*,ERR=800,END=800)UPPER
                        IF(UPPER.GT.250.)UPPER=250.
                   IF(LOWER.GT.UPPER)GOTO 800
               RANGE=(ABS((UPPER-LOWER)*.1))
820      YC(4)=LOWER+(     (RANGE*(ISEN-1)))
                   IF(YC(4).GT.UPPER)GOTO 2620
C        IF(YC(4).GT.100..OR.YC(4).LE.0.)GOTO 2620
         SENVAR=YC(4)
```

```
      GOTO 2440
900   CONTINUE
            IF(ISEN.GT.1)GOTO 920
      AREA1=GUIDE
      AREA2=ACCUR
      AREA3=UNDRL
            PRINT 910
910   FORMAT(' INPUT THE VALUES TO VARY GUIDANCE IMPACT ERROR BETWEEN.'
     *,/,' WHAT IS THE LOWER VALUE?')
            READ(IN,*,ERR=900,END=900)LOWER
               IF(LOWER.LE.0)LOWER=.01
            PRINT 915
915   FORMAT(' WHAT IS THE UPPER VALUE?')
            READ(IN,*,ERR=900,END=900)UPPER
                        IF(UPPER.GT.250.)UPPER=250.
                   IF(LOWER.GT.UPPER)GOTO 900
               RANGE=(ABS((UPPER-LOWER)*.1))
920      YC(5)=LOWER+(    (RANGE*(ISEN-1)))
                   IF(YC(5).GT.UPPER)GOTO 2620
         SENVAR=YC(5)
      GOTO 2440
1000  CONTINUE
            IF(ISEN.GT.1)GOTO 1020
      AREA1=MOTOR
      AREA2=QUAL
      AREA3=TEST
            PRINT 1010
1010  FORMAT(' INPUT THE VALUES TO VARY MOTOR QUAL TESTS BETWEEN.'
     *,/,' WHAT IS THE LOWER NUMBER OF TESTS; >20 ?')
            READ(IN,*,ERR=1000,END=1000)LOWER
            IF(LOWER.LT.20.)LOWER=20.
            PRINT 1015
1015  FORMAT(' WHAT IS THE UPPER NUMBER; <40 ?')
            READ(IN,*,ERR=1000,END=1000)UPPER
            IF(UPPER.GT.40.)UPPER=40.
                   IF(LOWER.GT.UPPER)GOTO 1000
               RANGE=(INT((UPPER-LOWER)*.1))
                   IF(RANGE.LT.1.)RANGE=1.
1020     QT(2)=LOWER+(INT(RANGE*(ISEN-1)))
               IF(QT(2).GT.UPPER)GOTO 2620
```

```
         SENVAR=QT(2)
      GOTO 2440
1100  CONTINUE
            IF(ISEN.GT.1)GOTO 1120
      AREA1=AIRFR
      AREA2=QUAL
      AREA3=TEST
            PRINT 1110
1110  FORMAT(' INPUT THE VALUES TO VARY AIRFRAME QUAL TESTS BETWEEN.'
     *,/,' WHAT IS THE LOWER NUMBER OF TESTS; >3 ?')
            READ(IN,*,ERR=1100,END=1100)LOWER
            IF(LOWER.LT.3.)LOWER=3.
            PRINT 1115
1115  FORMAT(' WHAT IS THE UPPER NUMBER; <9 ?')
            READ(IN,*,ERR=1100,END=1100)UPPER
            IF(UPPER.GT.9.)UPPER=9.
                   IF(LOWER.GT.UPPER)GOTO 1100
               RANGE=(INT((UPPER-LOWER)*.1))
                   IF(RANGE.LT.1.)RANGE=1.
1120     QT(3)=LOWER+(INT(RANGE*(ISEN-1)))
               IF(QT(3).GT.UPPER)GOTO 2620
         SENVAR=QT(3)
      GOTO 2440
1200  CONTINUE
            IF(ISEN.GT.1)GOTO 1220
      AREA1=LAUNC
      AREA2=QUAL
      AREA3=TEST
            PRINT 1210
1210  FORMAT(' INPUT THE VALUES TO VARY LAUNCHER QUAL TESTS BETWEEN.'
     *,/,' WHAT IS THE LOWER NUMBER OF TESTS; >2 ?')
            READ(IN,*,ERR=1200,END=1200)LOWER
            IF(LOWER.LT.2.)LOWER=2.
            PRINT 1215
1215  FORMAT(' WHAT IS THE UPPER NUMBER; <6 ?')
            READ(IN,*,ERR=1200,END=1200)UPPER
            IF(UPPER.GT.6.)UPPER=6.
                   IF(LOWER.GT.UPPER)GOTO 1200
               RANGE=(INT((UPPER-LOWER)*.1))
                   IF(RANGE.LT.1.)RANGE=1.
```

```
1220       QT(6)=LOWER+(INT(RANGE*(ISEN-1)))
                 IF(QT(6).GT.UPPER)GOTO 2620
           SENVAR=QT(6)
        GOTO 2440
1300    CONTINUE
              IF(ISEN.GT.1)GOTO 1320
        AREA1=FIREC
        AREA2=QUAL
        AREA3=TEST
              PRINT 1310
1310    FORMAT(' INPUT THE VALUES TO VARY FIRE CONTROL QUAL TESTS BETWEEN'
       *,/,' WHAT IS THE LOWER NUMBER OF TESTS; >2 ?')
              READ(IN,*,ERR=1300,END=1300)LOWER
              IF(LOWER.LT.2.) LOWER=2.
              PRINT 1315
1315    FORMAT(' WHAT IS THE UPPER NUMBER; <4 ?')
              READ(IN,*,ERR=1300,END=1300)UPPER
              IF(UPPER.GT.4.) UPPER=4.
                     IF(LOWER.GT.UPPER)GOTO 1300
                 RANGE=(INT((UPPER-LOWER)*.1))
                     IF(RANGE.LT.1.)RANGE=1.
1320       QT(4)=LOWER+(INT(RANGE*(ISEN-1)))
                 IF(QT(4).GT.UPPER)GOTO 2620
           SENVAR=QT(4)
        GOTO 2440
1400    CONTINUE
              IF(ISEN.GT.1)GOTO 1420
        AREA1=GUIDE
        AREA2=QUAL
        AREA3=TEST
              PRINT 1410
1410    FORMAT(' INPUT THE VALUES TO VARY GUIDANCE QUAL TESTS BETWEEN.'
       *,/,' WHAT IS THE LOWER NUMBER OF TESTS; >3 ?')
              READ(IN,*,ERR=1400,END=1400)LOWER
              IF(LOWER.LT.3.) LOWER=3.
              PRINT 1415
1415    FORMAT(' WHAT IS THE UPPER NUMBER; <9 ?')
              READ(IN,*,ERR=1400,END=1400)UPPER
              IF(UPPER.LT.9.) UPPER=9.
                     IF(LOWER.GT.UPPER)GOTO 1400
```

```
            RANGE=(INT((UPPER-LOWER)*.1))
                IF(RANGE.LT.1.)RANGE=1.
1420     QT(5)=LOWER+(INT(RANGE*(ISEN-1)))
            IF(QT(5).GT.UPPER)GOTO 2620
         SENVAR=QT(5)
      GOTO 2440
1500  CONTINUE
            IF(ISEN.GT.1)GOTO 1520
      AREA1=FLIGHT
      AREA2=TEST
      AREA3=UNDRL
            PRINT 1510
1510  FORMAT(' INPUT THE VALUES TO VARY FLIGHT TESTS BETWEEN'
     *,/,' WHAT IS THE LOWER NUMBER OF TESTS: >10 ?')
            READ(IN,*,ERR=1500,END=1500)LOWER
            IF(LOWER.LT.10.)LOWER=10.
            PRINT 1515
1515  FORMAT(' WHAT IS THE UPPER NUMBER OF TESTS; <25 ?')
            READ(IN,*,ERR=1500,END=1500)UPPER
            IF(UPPER.GT.25.)UPPER=25.
                IF(LOWER.GT.UPPER)GOTO 1500
            RANGE=(ABS(UPPER-LOWER))*.1
                IF(RANGE.LT.1.)RANGE=1.
1520     QT(1)=LOWER+(INT(RANGE*(ISEN-1)))
            IF(QT(1).GT.UPPER)GOTO 2620
         SENVAR=QT(1)
      GOTO 2440
1600  CONTINUE
            IF(ISEN.GT.1)GOTO 1620
      AREA1=DEVEL
      AREA2=COST
            PRINT 1625
1625  FORMAT(' DO YOU WANT TO VARY THE MAXIMUM OR MINIMUM DEVELOPMENT CC
     *ST?',/,' ENTER EITHER MAXIMUM OR MINIMUM.')
            READ(IN,1630,ERR=1600,END=1600)LABEL
      AREA3=LABEL
1630  FORMAT(A5)
            PRINT 1610,LABEL
1610  FORMAT(' INPUT THE LOWER VALUE FOR ',A8,' DEVELOPMENT COST. ')
            READ(IN,*,ERR=1600,END=1600)LOWER
```

```
                IF(LOWER.LE.0)LOWER=.01
             PRINT 1615,LABEL
1615   FORMAT(' WHAT IS THE UPPER COST FOR ',A8,'?')
             READ(IN,*,ERR=1600,END=1600)UPPER
             IF(UPPER.GT.125.)UPPER=125.
                    IF(LOWER.GT.UPPER)GOTO 1600
                RANGE=(ABS(UPPER-LOWER))
1620      SENVAR=LOWER+RANGE*.1*(ISEN-1)
          IF(ISEN.EQ.1.AND.SENVAR.EQ.0.)PRINT 2535
          IF(SENVAR.EQ.0.)GOTO 2600
          IF(LABEL.EQ.MAX)CDMAX=SENVAR
          IF(LABEL.EQ.MIN)CDMIN=SENVAR
       GOTO 2440
1700   CONTINUE
             IF(ISEN.NE.1)GOTO 1720
       AREA1=DEVEL
       AREA2=COST
       AREA3=INCEN
             PRINT 1710
1710   FORMAT(' INPUT THE RANGE TO VARY INCENTIVE % OF DEVELOPMENT COST?'
      *,/,' WHAT IS THE LOWER VALUE?')
             READ(IN,*,ERR=1700,END=1700)LOWER
                IF(LOWER.LE.0)LOWER=.01
             PRINT 1715
1715   FORMAT(' WHAT IS THE UPPER VALUE?')
             READ(IN,*,ERR=1700,END=1700)UPPER
                    IF(UPPER.GT.100.)UPPER=100.
                    IF(LOWER.GT.UPPER)GOTO 1700
1720   FCDMIN=LOWER+(ABS(UPPER-LOWER))*.1*(ISEN-1)
          IF(FCDMIN.LE.0.)PRINT 2535
          IF(FCDMIN.LE.0.)GOTO 2600
          DELPER=FCDMIN+FTDMIN+FRSMAX+FESMIN
          SENVAR=FCDMIN
          IF(DELPER.LE.15.)GOTO 2440
          RATIO2=(15.-FCDMIN)/(FTDMIN+FRSMAX+FESMIN)
          FTDMIN=FTDMIN*RATIO2
          FRSMAX=FRSMAX*RATIO2
          FESMIN=FESMIN*RATIO2
       GOTO 2440
1800   CONTINUE
```

```
            IF(ISEN.GT.1)GOTO 1820
      AREA1=FLIGHT
      AREA2=TESTC
            PRINT 1825
1825  FORMAT(' DO YOU WANT TO VARY THE EARLIEST OR LATEST FLIGHT TEST'
     *,/,' COMPLETION DATE? ENTER EITHER LATE OR EARLY.')
            READ(IN,1630,ERR=1800,END=1800)LABEL
      AREA3=LABEL
            PRINT 1810,LABEL
1810  FORMAT(' INPUT THE RANGE TO CHANGE ',A8,' COMPLETION DATE.'
     *,/,' WHAT IS THE LOWER VALUE?')
            READ(IN,*,ERR=1800,END=1800)LOWER
               IF(LOWER.LE.0)LOWER=.01
            PRINT 1815,LABEL
1815  FORMAT(' WHAT IS THE UPPER VALUE FOR ',A8,' COMPLETION?')
            READ(IN,*,ERR=1800,END=1800)UPPER
            IF(UPPER.GT.300.)UPPER=300.
                   IF(LOWER.GT.UPPER)GOTO 1800
               RANGE=(ABS(UPPER-LOWER))*.1
                   IF(RANGE.LT.1.)RANGE=1.
1820     SENVAR=LOWER+(INT(RANGE*(ISEN-1)))
               IF(SENVAR.GT.UPPER)GOTO 2620
         IF(LABEL.EQ.LATE)TDMAX=SENVAR
         IF(LABEL.EQ.EARLY)TDMIN=SENVAR
      GOTO 2440
1900  CONTINUE
            IF(ISEN.NE.1)GOTO 1920
      AREA1=FLIGHT
      AREA2=TESTC
      AREA3=INCEN
            PRINT 1910
1910  FORMAT(' INPUT THE UPPER VALUE FOR DELIVERY INCENTIVE.')
            READ(IN,*,ERR=1900,END=1900)UPPER
                   IF(UPPER.GT.100.)UPPER=100.
            PRINT 1915
1915  FORMAT(' WHAT IS THE LOWER VALUE FOR DELIVERY INCENTIVE?')
            READ(IN,*,ERR=1900,END=1900)LOWER
               IF(LOWER.LE.0)LOWER=.01
                   IF(LOWER.GT.UPPER)GOTO 1900
1920     FIDMIN=LOWER+(ABS(UPPER-LOWER))*.1*(ISEN-1)
```

```
          IF(ISEN.EQ.1.AND.FTDMIN.LE.0.)PRINT 2535
          IF(FTDMIN.LE.0.)GOTO 2600
          DELPER=FCDMIN+FTDMIN+FRSMAX+FESMIN
          SENVAR=FTDMIN
          IF(DELPER.LE.15.)GOTO 2440
          RATIO2=(15.-FTDMIN)/(FCDMIN+FRSMAX+FESMIN)
          FCDMIN=FCDMIN*RATIO2
          FRSMAX=FRSMAX*RATIO2
          FESMIN=FESMIN*RATIO2
        GOTO 2440
 2000   CONTINUE
              IF(ISEN.NE.1)GOTO 2020
        AREA1=RELIA
              PRINT 2005
 2005   FORMAT(' DO YOU WANT TO VARY THE UPPER OR LOWER LEVEL OF RELIABILI
       *Y?',/,' INPUT HIGH OR LOW.')
              READ(IN,1630,ERR=2000,END=2000)LABEL
        AREA2=LABEL
        AREA3=UNDRL
              PRINT 2010,LABEL
 2010   FORMAT(' WHAT IS THE UPPER VALUE FOR ',A8,' RELIABILITY?')
              READ(IN,*,ERR=2000,END=2000)UPPER
                   IF(UPPER.GT.100.)UPPER=100.
              PRINT 2015,LABEL
 2015   FORMAT(' WHAT IS THE LOWER VALUE FOR ',A8,' RELIABILITY?')
              READ(IN,*,ERR=2000,END=2000)LOWER
                 IF(LOWER.LE.0)LOWER=.01
                   IF(LOWER.GT.UPPER)GOTO 2000
 2020     SENVAR=LOWER+(ABS(UPPER-LOWER))*.1*(ISEN-1)
          IF(ISEN.EQ.1.AND.(SENVAR.GT.100..OR.SENVAR.LT.0.))PRINT 2535
          IF(SENVAR.GT.100..OR.SENVAR.LT.0.)GOTO 2440
          IF(LABEL.EQ.HIGH)RSMAX=SENVAR
          IF(LABEL.EQ.LOW)RSMIN=SENVAR
        GOTO 2440
 2100   CONTINUE
              IF(ISEN.NE.1)GOTO 2120
        AREA1=RELIA
        AREA2=INCEN
        AREA3=UNDRL
              PRINT 2110
```

```
2110  FORMAT(' INPUT THE UPPER VALUE OF INCENTIVE % FOR RELIABILITY.')
            READ(IN,*,ERR=2100,END=2100)UPPER
                    IF(UPPER.GT.100.)UPPER=100.
            PRINT 2115
2115  FORMAT(' WHAT IS THE LOWER VALUE?')
            READ(IN,*,ERR=2100,END=2100)LOWER
               IF(LOWER.LE.0)LOWER=.01
                    IF(LOWER.GT.UPPER)GOTO 2100
2120     FRSMAX=LOWER+(ABS(UPPER-LOWER))*.1*(ISEN-1)
         DELPER=FCDMIN+FTDMIN+FRSMAX+FESMIN
         SENVAR=FRSMAX
         IF(FRSMAX.LE.0.)PRINT 2535
         IF(FRSMAX.LE.0.)GOTO 2600
         IF(DELPER.LE.15.)GOTO 2440
         RATIO2=(15.-FRSMAX)/(FCDMIN+FRSMAX+FESMIN)
         FCDMIN=FCDMIN*RATIO2
         FTDMIN=FTDMIN*RATIO2
         FESMIN=FESMIN*RATIO2
      GOTO 2440
2200  CONTINUE
            IF(ISEN.NE.1)GOTO 2220
      AREA1=FIREC
            PRINT 2210
2210  FORMAT(' DO WANT TO CHANGE THE GREATER OR LESSER ACCURACY?',/,
     *' INPUT EITHER GREATER OR LESSER.')
            READ(IN,1630,ERR=2200,END=2200)
      AREA2=LABEL
      AREA3=UNDRL
            PRINT 2215,LABEL
2215  FORMAT(' WHAT IS THE UPPER VALUE FOR ',A8,' ACCURACY?')
            READ(IN,*,ERR=2200,END=2200)UPPER
            IF(UPPER.GT.300.)UPPER=300.
            PRINT 2216,LABEL
2216  FORMAT(' WHAT IS THE LOWER VALUE FOR ',A8,' ACCURACY?')
            READ(IN,*,ERR=2200,END=2200)LOWER
               IF(LOWER.LE.0)LOWER=.01
                    IF(LOWER.GT.UPPER)GOTO 2200
2220     SENVAR=LOWER+(ABS(UPPER-LOWER))*.1*(ISEN-1)
         IF(SENVAR.LT.0.)GOTO 2620
         IF(LABEL.EQ.GREAT)ESMAX=SENVAR
```

```
          IF(LABEL.EQ.IOW)ESMIN=SENVAR
        GOTO 2440
  2300  CONTINUE
              IF(ISEN.NE.1)GOTO 2320
        AREA1=FIREC
        AREA2=INCEN
        AREA3=UNDRL
              PRINT 2310
  2310  FORMAT(' INPUT THE RANGE TO VARY INCENTIVE % FOR ACCURACY .',/,
       *' WHAT IS THE UPPER VALUE?')
              READ(IN,*,ERR=2300,END=2300)UPPER
                    IF(UPPER.GT.100.)UPPER=100.
              PRINT 2315
  2315  FORMAT(' WHAT IS THE LOWER VALUE?')
              READ(IN,*,ERR=2300,END=2300) LOWER
                IF(LOWER.LE.0)LOWER=.01
                    IF(LOWER.GT.UPPER)GOTO 2300
  2320    FESMIN=LOWER+(ABS(UPPER-LOWER))*.1*(ISEN-1)
          DELPER=FCDMIN+FTDMIN+FRSMAX+FESMIN
          SENVAR=FESMIN
          IF(ISEN.EQ.1.AND.FESMIN.LE.0)PRINT 2535
          IF(FESMIN.LE.0)GOTO 2600
          IF(DELPER.LE.15.)GOTO 2440
          RATIO2=(15-FESMIN)/(FCDMIN+FRSMAX+FESMIN)
          FCDMIN=FCDMIN*RATIO2
          FTDMIN=FTDMIN*RATIO2
          FRSMAX=FRSMAX*RATIO2
        GOTO 2440
  2400  CONTINUE
              IF(ISEN.NE.1)GOTO 2420
        AREA1=DELIV
        AREA2=BLK
        AREA3=UNDRL
              PRINT 2410
  2410  FORMAT(' INPUT THE RANGE TO VARY DELIVERY DATE.',/,' WHAT IS THE E
       *ARLIEST DATE?')
              READ(IN,*,ERR=2400,END=2400)LOWER
                IF(LOWER.LE.0)LOWER=.01
              PRINT 2415
  2415  FORMAT(' WHAT IS THE LATER DATE?')
```

```
            READ(IN,*,ERR=2400,END=2400)UPPER
                 IF(LOWER.GT.UPPER)GOTO 2400
              RANGE=(ABS(UPPER-LOWER))*.1
                 IF(RANGE.LT.1.)RANGE=1.
 2420     DD(13)=LOWER+(INT(RANGE*(ISEN-1)))
              IF(DD(13).GT.UPPER)GOTO 2620
          IF(DD(13).LE.0.AND.ISEN.EQ.1)PRINT 2535
          IF(DD(13).LE.0.)GOTO 2600
          SENVAR=DD(13)
 2440   CONTINUE
          CALL INPUT3
          CALL PRESET
          CALL TRADE
          CALL DESRES
              GOTO(2445,2450),KKPP
 2445     CALL DESRET
 2450     CALL SE
              IF (NTEMP(1) .EQ. 3 .OR. NTEMP(1) .EQ. IDP) GO TO 130
            ITCT=0
          DO 110 M=1,INA
            IF (QTY(M)-0.1) 110,110,115
 115        IF(ITERM(M)-IDP-1) 110,120,110
 120        ITCT=ITCT+1
 110      CONTINUE
               GO TO 140
 130        IDP=7
            ITCT=-1
            ITAB(25)=IDP
C
 140      CALL MOD9
                                    IF(IVER2.EQ.KDP)GOTO 150
                                    IF(IVER2.NE.KDP-1)GOTO 155
          CALL RERUN
 150    CONTINUE
 155      CALL REPORT
          CALL PERF
                                    CALL GET(KDP)
        IF(SENSTY.EQ.1.0 )CTC(8)=SENVAR
        IF(SENSTY.EQ.2.0 )CTC(7)=SENVAR
        IF(SENSTY.EQ.3.0 )CTC(5)=SENVAR
```

```
      IF (SENSTY.EQ.4.0 )YC(1)  =SENVAR
      IF (SENSTY.EQ.5.0 )YC(2)  =SENVAR
      IF (SENSTY.EQ.6.0 )YC(3)  =SENVAR
      IF (SENSTY.EQ.7.0 )YC(4)  =SENVAR
      IF (SENSTY.EQ.8.0 )YC(5)  =SENVAR
      IF (SENSTY.EQ.9.0 )QT(2)  =SENVAR
      IF (SENSTY.EQ.10.0)QT(3)  =SENVAR
      IF (SENSTY.EQ.11.0)QT(6)  =SENVAR
      IF (SENSTY.EQ.12.0)QT(4)  =SENVAR
      IF (SENSTY.EQ.13.0)QT(5)  =SENVAR
      IF (SENSTY.EQ.14.0)QT(1)  =SENVAR
      IF (SENSTY.EQ.16.0)FCDMIN=SENVAR
      IF (SENSTY.EQ.18.0)FTDMIN=SENVAR
      IF (SENSTY.EQ.20.0)FRSMAX=SENVAR
      IF (SENSTY.EQ.22.0)FESMIN=SENVAR
      IF (SENSTY.EQ.23.0)DD(13) =SENVAR
      IF (LABEL.EQ.MAX)CDMAX=SENVAR
      IF (LABEL.EQ.MIN)CDMIN=SENVAR
      IF (LABEL.EQ.LATE)TDMAX=SENVAR
      IF (LABEL.EQ.EARLY)TDMIN=SENVAR
      IF (LABEL.EQ.HIGH)RSMAX=SENVAR
      IF (LABEL.EQ.LOW.AND.SENSTY.EQ.19.0)RSMIN=SENVAR
      IF (LABEL.EQ.GREAT)ESMAX=SENVAR
      IF (LABEL.EQ.LOW.AND.SENSTY.EQ.21.0)ESMIN=SENVAR
C THE FOLLOWING LINES ARE REPRODUCED FROM PROSUM; PROSUM COULD NOT BE
C CALLED DO TO THE PRINT REQUIREMENTS.
      TMTABA=0.
      TMTABB=0.
      CTESTA=CTEST(1)-(CT(19)+CT(20))
      FCTOH=CTESTA/CTEST(1)
      CTOH(1)=CTOH(1)*FCTOH
      Y(1)=CTEST(1)+CTOH(1)
      DO 5 J=1,10
      ITAB(J)=TAB(19,J,1)
    5 CONTINUE
      DO 3310 M=1,INA
C CHANGED LABELS 10 AND 15, TO 3310 AND 3315, DO TO CONFLICT GENERATED
C IN BRINGING THIS ROUTINE OUT OF SUBROUTINE PROSUM.
      IF(IQ(M)-1)3310,3315,3310
3310  CONTINUE
```

```
C
      M=INA
3315  Y(2)=TFIN(M)
   25 Y(3)=TAB(14,16,2)
      Y(4) = TAB(14,20,2)
C         . . . ROW 19 CHANGED TO ROW 20  1970 APRIL 1 . . .
   30 K=4
      M=N(K)
      Y(4)=Y(4)*YF(M)
      Y(5)=C10
C
      PER(1)=CDMAX
      AN(1)=CDMIN
      FYC(1)=FCDMIN
      PER(2)=TDMAX
      AN(2)=TDMIN
      FYC(2)=FTDMIN
      AN(3)=RSMAX
      PER(3)=RSMIN
      FYC(3)=FRSMAX
      PER(4)=ESMAX
      AN(4)=ESMIN
      FYC(4)=FESMIN
      PER(5)=CPUMAX
      AN(5)=CPUMIN
      FYC(5)=FCPMIN
      SEVA=0.0
      FRATIC(8)=0.0
      Y(1)=Y(1)/1000000.0
      Y(3)=Y(3)*100.0
C
C THIS SET OF STEPS TC LABLE 50 CHECKS THE MOST DESIREAELE VALUES,
C AN(), VS ACHIEVED VALUES, Y(), AND LEAST DESIREABLE VALUE, PER()
C VS ACHIEVED VALUE IN ORDER TO PLACE THE FEE WITHIN THE LINEAR RATIO
C BCUNDS OF THE DESIRED FEE.
              DO 50 I=1,5
                 IF(I.EQ.3)GOTO 48
      IF(Y(I)-AN(I).GE.0..AND.PER(I)-Y(I).GE.0.AND.PER(I).NE.AN(I))
     *FRATIO(I)=FYC(I)*(ABS((PER(I)-Y(I))/(PER(I)-AN(I))))
      IF(Y(I)-PER(I).GE.0.)FRATIO(I)=0.0
```

```
               IF(AN(I)-Y(I).GT.0.) FRATIO(I)=FYC(I)
                    GOTO 49
48    IF(Y(3)-PER(3).GE.0..AND.AN(3)-Y(3).GE.0..AND.AN(3).NE.PER(3))
     *FRATIO(3)=FYC(3)*(ABS((Y(3)-PER(3))/(AN(3)-PER(3))))
               IF(PER(3)-Y(3).GT.0.)FRATIO(3)=0.0
               IF(Y(3)-AN(3).GE.0.)FRATIO(3)=FYC(3)
49             YFEE(I)=FRATIO(I)*(CDMAX+CDMIN)/200.0
               SPVA=SPVA+YFEE(I)
               FRATIO(8)=FRATIO(8)+FRATIO(I)
50          CONTINUE
      TOC=Y(1)+SPVA+DECOST/1000000.0
C
C CHANGED IY TO IY2 TO DIFFERENTIATE FROM IY4 BELOW. 2/84
      IY2= Y(2)
C
      Y(3) = TAB(14,16,2) * 100.
C
C CHANGED IY TO IY4 TO DIFFERENTIATE FROM IY2 ABOVE. 2/84
      IY4=TAB(14,20,2)*100.
C        . . . IY=Y(4) CHANGED TO ABOVE  1970 APRIL 1 . . .
      CPUMAX = CPUMAX / 1000.
      CPUMIN = CPUMIN / 1000.
            IF(ISEN.EQ.1)CALL FRTCMS ('CLRSCRN ')
            IF(ISEN.EQ.1)PRINT 2535,AREA1
            IF(ISEN.EQ.1)PRINT 2536,AREA2,AREA3
               CALL FSEFF(TOC,EI)
               EI=ABS(EI)
2535  FORMAT(1X,69(' '),/,1X,
     *'#|SENSITIV|DEVELOPMNT|FLIGHTTEST|RELIABILTY|ACCURACY|FEE|CONTRACT
     *|EI|',/,1X,
     *' |',A8,  '|FEE| COST |FEE| DATE |FEE|  %   |FEE|YDS | % |  COST
     *|  |')
2536  FORMAT(1X,' |',A8,
     *          '| % |       | % |COMPLT| % |       | % |    |TCT|   $M
     *|  |',/,1X,
     *_|',A8,  '|_ACHIEVED_|_ACHIEVED_|_ACHIEVED_|ACHIEVED|ACH|ACHIEVED
     *|__|')
            J=ISEN-1

            PRINT 2525,J,SENVAR,FRATIO(1),Y(1),FRATIO(2),IY2,FRATIO(3),
```

```
     *Y(3),FRATIO(4),IY4,FRATIO(8),TOC,EI
2525  FORMAT(1X,I2,1X,F6.2,F6.2,F6.2,F5.2,I4,F6.2,F6.2
     *,F5.2,I4,F6.2,F6.2,F6.2)
                SAVDC(1,ISEN)=ABS(FRATIO(1))
                SAVDC(3,ISEN)=ABS(FRATIO(2))
                SAVDC(5,ISEN)=ABS(FRATIO(3))
                SAVDC(7,ISEN)=ABS(FRATIO(4))
                SAVDC(2,ISEN)=ABS(Y(1))
                SAVDC(4,ISEN)=ABS(Y(2))
                ISAVD2(ISEN)=IY2
                SAVDC(6,ISEN)=ABS(Y(3))
                SAVDC(8,ISEN)=ABS(Y(4))
                ISAVD4(ISEN)=IY4
                SAVDC(9,ISEN)=ABS(FRATIO(8))
                SAVDC(10,ISEN)=ABS(TOC)
                SAVDC(11,ISEN)=ABS(EI)
                SAVSEN(ISEN)=SENVAR
             IF(ISEN.EQ.1) EIOPT=ABS(EI)
             IF(ISEN.EQ.1) SELX=1.
             IF(EI.GT.EIOPT.AND.SAVSEN(ISEN).GT.0.) SELX=ISEN
             IF(EI.GT.EIOPT.AND.SAVSEN(ISEN).GT.0.) EIOPT=ABS(EI)
2600                                          CONTINUE
2620                                          CONTINUE
                             PAUSE ';PRESS <<ENTER>> TO CONTINUE.'
                                              RETURN
      ENTRY PLOT(CKPLT)
                                              CKPLT=ISEN
                                              RETURN
      ENTRY PLTNCT(CKPLT)
                                              ISEN=CKPLT
                                              RETURN
      ENTRY SNSPRT
                             CALL FRTCMS('CLRSCRN ')
                             PRINT 2535,AREA1
                             PRINT 2536,AREA2,AREA3
                       DO 3995 JITIG=1,11
                                      JITIGA=JITIG-1
             PRINT 2525,JITIGA,SAVSEN(JITIG),SAVDC(1,JITIG),SAVDC(2,JIT
     *IG),SAVDC(3,JITIG),ISAVD2(JITIG),SAVDC(5,JITIG),SAVDC(6,JITIG),SAV
     *DC(7,JITIG),ISAVD4(JITIG),SAVDC(9,JITIG),SAVDC(10,JITIG),SAVDC(11,
```

```
     *JITIG)
3995      CONTINUE
                                 PAUSE ';PRESS <<ENTER>> TO CONTINUE.'
                                                RETURN
      ENTRY OPTIM
                IF(SAVSEN(SELX).LE.0.0) RETURN
          CALL GET(KDP)
4005  GOTO(4010,4020,4030,4040,4050,4060,4070,4080,4090,4100,4110,4120,
     *4130,4140,4150,4160,4170,4180,4190,4200,4210,4220,4230)
     *,SENSTY
4010      CONTINUE
                CTC(8)=SAVSEN(SELX)
                     GCTO 4400
4020      CONTINUE
                CTC(7)=SAVSEN(SELX)
                     GOTO 4400
4030      CONTINUE
                CTC(5)=SAVSEN(SELX)
                     GOTO 4400
4040      CONTINUE
                 YC(1)=SAVSEN(SELX)
                     GOTO 4400
4050      CONTINUE
                 YC(2)=SAVSEN(SELX)
                     GCTO 4400
4060      CONTINUE
                 YC(3)=SAVSEN(SELX)
                     GOTO 4400
4070      CONTINUE
                 YC(4)=SAVSEN(SELX)
                     GOTO 4400
4080      CONTINUE
                 YC(5)=SAVSEN(SELX)
                     GOTO 4400
4090      CONTINUE
                 CT(2)=SAVSEN(SELX)
                     GOTO 4400
4100      CONTINUE
                 CT(3)=SAVSEN(SELX)
                     GOTO 4400
```

```
4110          CONTINUE
                      QT(6)=SAVSEN(SELX)
                            GOTO 4400
4120          CONTINUE
                      QT(4)=SAVSEN(SELX)
                            GOTO 4400
4130          CONTINUE
                      QT(5)=SAVSEN(SELX)
                            GOTO 4400
4140          CONTINUE
                      QT(1)=SAVSEN(SELX)
                            GOTO 4400
4150          CONTINUE
        IF(LABEL.EQ.MAX)CDMAX=SAVSEN(SELX)
        IF(LABEL.EQ.MIN)CDMIN=SAVSEN(SELX)
                            GOTO 4400
4160          CONTINUE
                     FCDMIN=SAVSEN(SELX)
                            GOTO 4400
4170          CONTINUE
        IF(LABEL.EQ.LATE) TDMAX=SAVSEN(SELX)
        IF(LABEL.EQ.EARLY)TDMIN=SAVSEN(SELX)
                            GOTO 4400
4180          CONTINUE
                     FTDMIN=SAVSEN(SELX)
                            GOTO 4400
4190          CONTINUE
        IF(LABEL.EQ.HIGH)RSMAX=SAVSEN(SELX)
        IF(LABEL.EQ.LOW)RSMIN=SAVSEN(SELX)
                            GOTO 4400
4200          CONTINUE
                     FRSMAX=SAVSEN(SELX)
                            GOTO 4400
4210          CONTINUE
        IF(LABEL.EQ.GREAT)ESMAX=SAVSEN(SELX)
        IF(LABEL.EQ.LOW)ESMIN=SAVSEN(SELX)
                            GOTO 4400
4220          CONTINUE
                    FESMIN=SAVSEN(SELX)
                            GOTO 4400
```

```
4230        CONTINUE
                  DI(13)=SAVSEN(SELX)
4400        CONTINUE
                        CALL SETUP(KDP)
      ENTRY PLOTST(ISEN)
      RETURN
      END
C***************************************************************************
C
      SUBROUTINE PRCTCK
C
      COMMON REST(48),TAB(20,25,3),CTEST(3),CTOH(3),XMIS(2),ISUBS(48)
     &   ,    IFAC(48),   NT(48),    IRC(49),    IDDT(48)
     &   ,    IFD(48),   ITERM(48),   QTY(48),   CMAT(48)
      COMMON   RESS(48),    RESU(48),    RESB(48),    ITYPE(48)
     &   ,    PERC(48),   TDUE(48),    DEC(3),    Y(8)
     &   ,    IPRC(48),   RESC(48),    YF(48),    RESA(48)
     &   ,    TSTAR(48),   CL(6,5),    ALN(48),    CT(48)
     &   ,    TFIN(48),   TMIN(48),    IQ(48),    CUN(30)
      COMMON   RESMIN(30),    RESMAX(30),    AK(30),    ISW(3)
     &   ,    ITAB(25),   NN(8),    N(8),    INDX(8)
     &   ,    RATIO(8),   AN(9),    ADJ(2,8),    PDR(6)
     &   ,    PER(6),   PVR(6),   CD,   ICD,   INA,INF,IDP
      COMMON CPUMAX,CPUMIN,FCPMIN,X,NTEMP(25),COV,C10,TRADET,TRADER,
     &TRADEA,DAM1,DAM2,DEF,SUR,COSTM,COSTD,COSTP1
      COMMON   COSTP2,   COSTOT,   ITCT,   ALPHA,   PRIOR
     &   ,    TRADEP,   TF,   WA,   IACT(48,2),   IPC(48,5)
     &   ,    TDS(48,2),   JMIN,   IC,   IFIG,   KDP,   MR
     &   ,    NC,ICOFG,KP,KKPP,ITEAM
      COMMON RSMAX,RSMIN,FRSMAX,ESMAX,ESMIN,FESMIN,CDMAX,CDMIN,
     &FCDMIN,TDMAX,TIMIN,FTDMIN,QT(25),YC(6),CTC(8),DD(32),MIS(9)
      INTEGER IN/5/
      REAL*4 ANS,CHG/'CHG '/,E/'E   '/,YES/'Y   '/
      REAL*4 NORM/'NORM'/,MORE/'MORE'/
      REAL*8 GREAT/'GREATER'/,LESS/'LESS'/,SIZE
C
400   PERTOT=FCDMIN+FTDMIN+FRSMAX+FESMIN
      IF(PERTOT.GT.14.999999.AND.PERTOT.LT.15.000001)RETURN
      CALL PRTCMS('CLRSCRN ')
      IF(PERTOT.GT.15.)SIZE=MORE
```

```
      IF(PERTOT.LT.15.)SIZE=LESS
      PRINT 5,SIZE,FCDMIN,FTDMIN,FRSMAX,FESMIN
5     FORMAT(' *****WARNING*****TOTAL OF FEE PERCENTAGES IS ',A8,' THAN
     *15%.',/,                                                 ' THE VALUES
     * YOU PREVIOUSLY SUBMITTED ARE: ',/,' COST FEE INCENTIVE=',F5.2,/,
     *' DELIVERY DATE FEE INCENTIVE=',F5.2,/,' RELIABILITY INCENTIVE FEE
     *=',F5.2,/,' ACCURACY INCENTIVE FEE=',F5.2,/,' REDO YOUR INPUT.',/)
500   PRINT 510
510   FORMAT(' ENTER THE COST INCENTIVE FEE.')
      CALL CHANGE(FCDMIN)
      PERTOT=FCDMIN
      IF(PERTOT.LE.15.)GOTO 600
      PRINT 605, PERTOT
605   FORMAT(' TOTAL OF INCENTIVE FEES (',F5.2,') EXCEEDS 15%.')
      CALL EXITS
      GOTO 400
600   DIFF=15.-PERTOT
      PRINT 615,DIFF
615   FORMAT(' YOU HAVE ',F5.2,'% OF THE INCENTIVE FEES REMAINING.')
      PRINT 515
515   FORMAT(' ENTER THE DELIVERY DATE INCENTIVE FEE.')
      CALL CHANGE(FTDMIN)
      PERTOT=FCDMIN+FTDMIN
      IF(PERTOT.LE.15.)GOTO 620
      PRINT 605, PERTOT
      CALL EXITS
      GOTO 600
620   DIFF=15.-PERTOT
      PRINT 615,DIFF
      PRINT 520
520   FORMAT(' ENTER THE RELIABILITY INCENTIVE FEE.')
      CALL CHANGE(FRSMAX)
      PERTOT=FCDMIN+FTDMIN+FRSMAX
      IF(PERTOT.LE.15.)GOTO 625
      PRINT 605, PERTOT
      CALL EXITS
      GOTO 620
625   DIFF=15.-PERTOT
      PRINT 615,DIFF
      PRINT 525
```

```
525   FORMAT(' ENTER THE ACCURACY INCENTIVE FEE.')
      CALL CHANGE(FESMIN)
      PERTOT=FCDMIN+FTDMIN+FRSMAX+FESMIN
      IF(PERTOT.EQ.15.)RETURN
      CALL EXITS
      GOTO 400
      END
```

APPENDIX H

PROJMNG PROGRAM VARIABLES GLOSSARY

AA	
AAA	
ACCUR	Character string for 'accuracy'
ADJ	Recurrsive TDU sum
ADUMP	Array variable for data file fields 6 through 33
AFSE	Average Force Structure Effectiveness
AIRFR	Character string for 'airframe'
AJJ	
AK	
ALN	
ALPHA	
AMPER	Character for curve plot;&
AN	Variable for answer to Yes/No questions
ANO	Negative answer to Yes/No questions
ANS	Variable for answer to Yes/No questions
ANT	Variable for answer to Yes/No questions
AREA1	Character buffer for creating labels in Sensitivity Table
AREA2	Character buffer for creating labels in Sensitivity Table
AREA3	Character buffer for creating labels in Sensitivity Table
AT	Character @
BIGGER	Character string used in Sensitivity Table label
BLK	Character string for '00000000'
CD	Development cost
CDMAX	Maximum development cost
CDMIN	Minimum development cost
CENT	Character for curve plot;¢
CHANGE	Subprogram for changing real variables
CHANGI	Subprogram for changing imaginary variables
CHARA	Character string buffer to put characters in plot lines
CHARKY	Array for character key characters
CHARKDF	Array for character key definitions

CKPIT	Subroutine to check the availability of plot data
CL	
CMAT	
CODE	Common buffer label
COLON	Character for plot curve label;:
COM	Character for plot curve label;,
COSTD	Cost of development
COSTM	Cost of manufacture
COSTCT	Cost total
COSTP1	Cost parallel contractor a
COSTP2	Cost parallel contractor b
COV	
CPUMAX	Cost of production maximum
CPUMIN	Cost of production minimum
CT	Cost of tests total
CTC	Input desired engineering costs
CTEST	Cost of system tests
CTOH	Cost of tests overhead
CUN	
C10	
D	
DAM1	
DAM2	
DASH	Character for curve plot;-
DATCOD	File for DATACODE TEAMxx
DATFIL	File for DATAFILE TEAMxx
DD	Input desired Delivery Date
DEC	Decrease for smallest performance factor impact
DECk	Decrease for smallest performance factor impact
DECmin	Decrease for smallest performance factor impact
DEF	
DELAY	
DELIV	Character string for 'delivery' in sensitivity Table
DELPER	
DESRES	Subroutine to decrease resources
DESRET	Subroutine to decrease resources because of tests
DEVEL	Character string for 'development' in Sensitivity Table
DFTE	
DFTR	
D	
DOL	Character $
DPCOST	Decision Point cost
DQT1	Differential value of flight tests

DQT2	Differential value of motor qual tests
DQT3	Differential value of airframe qual te
DQT4	Differential value of fire control qua
DQT5	Differential value of guidance qual te
DQT6	Differential value of launcher qual te
DRSMAX	Differential value applied to maximum reliability
E	
EARLY	Character string for determining sensitivity analysis selection
EI	Effectiveness index
EIOPT	Optimum effectiveness index
ENGIN	Character string for Sensitivity Table 'engineering'
ERRSET	Macro subroutine for resetting error functions on the IBM
ESMAX	Accuracy error maximum
ESMIN	Accuracy error minimum
EXITS	Subroutine to choose to stop or reloop back into the program
EXP	
F	
FCDMIN	Fee for development cost incentive
FCPMIN	Fee for production cost incentive
FCTOH	
FEETCT	Total of fees
FESMIN	Fee for accuracy incentive
FINAL	Label for team data files to identify final DF proposal
FINISH	Subroutine for submitting final and proposal contract proposal
FIREC	Character string for Sensitivity Table 'fire control'
FLAG	Flag for generating variation to the Selection Table
FLIGHT	Character string for the Sensitivity Table 'flight'
FL21	Flag for the monitors sessions
FRSMAX	Fee for reliability incentive
FRTCMS	Macro subroutine for clearing the screen in the IBM
FSE	Force structure effectiveness
FSEFF	Subroutine for force structure effectiv
FTDMIN	Fee for test completion date incentive
FUNDS	Character string for Sensitivity Table label 'funds'
G	
GET	Subroutine to place data file buffer data in array buffer
GREAT	Sensitivity selection label for parameter choice

IRC	
ISAVD2	save sensitivity data for flight test date
ISAVD4	save sensitivity data for accuracy
ISEN	for do loop of 10 iterations in Sensitivity Table
ISUBS	
ISW	
ITAB	
ITCT	
ITEM	Team number
ITEAMB	Team number
ITEAM	Team number
ITERM	
ITYPE	
IVER	DP loops of the main program
IVER2	Do loops of the Sensitivity achieved value loop
IXBASE	Integer value for plot x-axis
IXNM	Integer x-axis minimum value
IYNM	Integer y-axis minimum value
IY2	Integer for flight test completion date
IY4	Integer for accuracy
J	Integer for do loops
J1	
J2	
J3	
JA	Integer for do loops
JDP	INTEGER FOR DP
JITIG	Integer for do loops
JITIGA	Integer for do loops
JJ	Integer for do loops
JMIN	Integer for do loop finding minimum axis values
K	Integer do loop variable
KA	
KB	
KDP	Current DP
KGO	Assigned goto
KGOA	Assigned goto
KGOB	Assigned goto
KIQ	
KKPP	Flag for data input at curent dp
KP	Flag for dtat input at DP-3
KTB	
KTR101	
KZ	
KZRW	
L	
LATE	Sensitivity selection parameter choice

LAUNC	Character string for Sensititvity Table
LESS	Character <
LINE	Plot line array
LINECL	Plot line array columne
LNCREM	Integer increment
LL	
LOW	Sensitivity selection parameter choice
LOWER	Character string for Sensitivity Table
M	
MAINT	Character string for Sensitivity Table
MEM	
MEMO	
MFTDCL	MINIMUM FLIGHT TEST COST
MINA	
MIS	
MISCT	
MK	
MOD9	Subroutine for achieved value calculation
MORE	Sensitivity selection parameter choice
MOTOR	Character string for Sensitivity Table
MR	Monitor mode selected
M6	
N	
NC	Contractor choise
NE	
NDUMP	Team data file first five fields
NEWDP	Newly selected DP level
NN	
NORM	
NR	
NSTTP	Table number from which to read data
NT	
NTEMP	Temporary buffer for DP
NULL	
OPTIM	Character string for Sensitivity Table 'optimum'
P	
PAGEIN	Subroutine for page input
PAGES	Variable for pages to be input
PAMAX	
PAMIN	
PARAI	
PD	
PDA	
PCTFIG	Flag for per plotting percentage plot
PDR	
PER	
PERC	
PERF	
PERTCT	Total percentage

GUIDE	Character string for Sensitivity Table 'guidance'
HIGH	Sensitivity selection labelfor parameter choice
HORIZ	Character for plot lines;
I	Integer variable for incrementing do loops
IACT	
IC	incremental costs overall
ICD	incremental development costs
ICODE	Security code buffer
ICOFG	Integer cost of guidance
IDDT	Integer delivery date
IDP	Initial DP computed for
IDUMP	Integer variable array for first five fields of team data file
IFAC	
IFAULT	Count for security code violations
IFD	
IFIG	
IKDP	Integer for do loop variable referring to DP as stored KDP
IN	Integer for do loop incrementing
INA	Initial accuracy from tables
INCEN	Character string for Sensitivity Table 'incentive'
INCOD	Security code input from the team violations
INDX	Index for selecting table variables
INF	
INPUT2	Subroutine for input of data to current DP
INPUT3	Subroutine for input of data from table files
INTEN	Proposal status
IPC	
IPG	
IPRC	
IQ	
IQUEPP	Integer for question response for percentage plot
IQUESE	question response: exit
IQUESM	question response: monitor's main menu
IQUESO	question response: Use optimum ei for parameter value
IQUESP	question response: sensitivity table plot
IQUESS	question response: sensitivity table print
IQUSPP	question response: variable axis plot

PLOT	
PLOTSI	Flag to set the plot capability on
PLOTVA	Common buffer for plot data
PLTNOT	Flag to cancel plot when database is changed
PLTPCT	Entry routine for percent plot
PLTSCH	Subroutine for plotting
PLTSCL	Entry routine for scaled plot
PM	
PMA	
PMAX	
PMIN	
PRCTCK	Subroutine for checking the percentage total
PRD	Character "."
PRESET	Subroutine
PRICE	
PROP	Character field for Proposal printout
PROSM2	
PROSUM	Subroutine to display achieved values
PRTPCT	Character "%"
PV	
PVR	
QQ	
QT	Quantity of tests
QTY	Adjusted quantity of tests
QUES	answer to selection of menu
QUOTE	
R	
RATIO	Achieved value extrapolation ratio
RATIC2	Achieved value extrapolation ratio
RBLK	Character string of blanks (real)
REPCRT	Subroutine for achieved value table calculations
RERUN	Subroutine for zeroizing recurrsive values in DP runs
RESA	Resource calculation
RESB	Resource calculation
RESC	Resource calculation
RESMAX	Resourse maximum input
RESMIN	Resource minimum input
RESS	Resource calculation
REST	Resource calculation
RESU	Resource calculation
RSMAX	Reliability maximum
RSMIN	Reliability minimum
S	
SAVDC	Sensitivity table data storing array
SAVSEN	Sensitivity parameter saved array
SCHEDU	Subroutine for determining schedule impact on EI

SE	Subroutine
SELECT	Subroutine to change Selection Menu.
SELX	Buffer for sensitivity parameter level w cpt ei
SEMICO	Character string for curve plot; ;
SENSTY	Choice from Sensitivity Menu
SENVAR	Sensitivity Table parameter value
SETUP	Subroutine to return data values to buffer
SIZE	Plot y-axis choice
SNSITV	Subroutine for sensitivity analysis
SNSPRT	Entry routine to reprint Sensitivity Table
SPEA	
SPER	
SPRIT	
START1	Date for start of block 1 production
STAT	Status of the team DP level
STORE	Subroutine to write data files onto the disk
STORED	Common buffer for database array
STORFL	Stored value of flag
SUB	
SUMFSE	Sum of force structure effectiveness
SUR	
T	
T1	
T10	
T11	
T2	
T3	
T4	
T5	
T6	
T9	
TAB	Array of data from the data tables
TDMAX	Flight test completion date desired max
TDMIN	Flight test completion date desired min
TDS	
TDUE	
TEAMIN	Subroutine to input team number and verify security code
TEAM01	Team 1
TEAM02	Team 2
TEAM03	Team 3
TEAM04	Team 4
TEAM05	Team 5
TEAM06	Team 6
TEAM07	Team 7
TEAM08	Team 8
TEAM09	Team 9

TEAM10	Team 10
TEAM11	Team 11
TEAM12	Team 12
TEAM13	Team 13
TEAM14	Team 14
TEAM15	Team 15
TEAM16	Team 16
TEAM17	Team 17
TEAM18	Team 18
TEAM19	Team 19
TEAM20	Team 20
TERM1	
TERM3	
TERM4	
TF	
TFIN	
TMIN	
TMTAEA	
TMTAEB	
TOC	Total cost of development
TRADE	Subroutine for trade offs impact
TRADEA	tradeoff value for accuracy
TRADEP	tradeoff value for performance
TRADER	tradeoff value for reliability
TRADET	tradeoff value for tests
TSTAR	
TTEST	
UNDRL	
UPPER	Sensitivity analysis parameter choice
USERID	Computer users computer serial number
VALUE	Character string for Sensitivity Table 'value'
VERT	Character for plot vertical lines
WA	
X	
X3	
XBASE	Plot x axis value array
XCTC	
XC7	
XC8	
XDECK	
XMIS	
XRAY	
XREST	
XRESU	
XXX	
XYCK	
XY1	
XY2	
XY3	

Y	
YC	
YCURVE	Curve data arrays
YDC	
YES	Answer to Yes/No questions
YF	
YFAC	Y-axis scale
YFACT	Y-axis scale
YMAX	Y-axis maximum value
YMIN	Y-axis minimum value
YFEE	Achieved value fee in dollars
YMN	Y-axis minimum
YMX	Y-axis maximum
YSCAIE	Y-axis scale
ZEBRA	Contractor name
ZERNUL	Character string for zero or null (blank)
ZERC	
ZRSU1	
ZRSU2	
ZRW	Subroutine to zeroize all non-data base equivalent values
ZTB1	
ZTB2	
ZTB3	
ZTB4	
ZTB5	
ZTB6	
ZTB7	
ZTB8	
ZULU	
ZWEEK	Final lot deployment week

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Defense Logistics Studies Information Exchange U.S. Army Logistics Management Center Fort Lee, Virginia 23801	1
3.	Academic Group Chairman, Code 39 Command, Control and Communications Group Naval Postgraduate School Monterey, California 93943	1
4.	Professor M. Kline, Code 54KX Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943	10
5.	Professor G. Bradley, Code 52BZ Department of Computer Science Naval Postgraduate School Monterey, California 93943	5
6.	LCDR J. Ferris, SC, USN, Code 54Fj Department of Administrative Sciences Naval Postgraduate School, Monterey, California 93943	1
7.	Dr. A. Singer Director Weapons Acquisition Simulation National Defense University Industrial College of the Armed Forces Fort McNair, Washington, D.C. 20319	2
8.	Dr. Hans Ebenfelt System AB Kungsgatan 8 S-111 43 Stockholm, Sweden	1
9.	Colonel Lui Pao Chen Director, Special Projects Organization Ministry of Defence Departure Building Paya Lebar Airport Singapore 1953	1
10.	Lieutenant Colonel Ralph Shain Army R&D Division Headquarters, Ministry of Defence Tel Aviv, Israel	1

11. Mr. Robert Butler 1
The Assessment Group
710 Wilshire Boulevard
Santa Monica, California 90401

12. Library, Code 0142 2
Naval Postgraduate School
Monterey, California 93943

13. LCDR G. Schultz, USN 2
1153 Sandwood Place
San Pedro, CA 90731

208101
Thesis
S36453
c.1

Schultz
PROJMNG FORTRAN: an
interactive computer
program for use with the
Defense Management Si-
mulation exercise.

15 FEB 87 32108
15 FEB 87 32108

208101
Thesis
S36453 Schultz
c.1
PROJMNG FORTRAN: an
interactive computer
program for use with the
Defense Management Si-
mulation exercise.

AMERICAN BINDERY
TOPEKA KANSAS
2

thesS36453
PROJMNG FORTRAN :
3 2768 001 00349 4
DUDLEY KNOX LIBRARY